Titre: Title:	Modèle de prévision des taux de clics des annonces textuelles sur les moteurs de recherche
Auteur: Author:	Farooq Sanni
Date:	2017
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Sanni, F. (2017). Modèle de prévision des taux de clics des annonces textuelles sur les moteurs de recherche [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/2725/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/2725/
Directeurs de recherche: Advisors:	Luc Adjengue
Programme: Program:	Maîtrise recherche en mathématiques appliquées

UNIVERSITÉ DE MONTRÉAL

MODÈLE DE PRÉVISION DES TAUX DE CLICS DES ANNONCES TEXTUELLES SUR LES MOTEURS DE RECHERCHE

FAROOQ SANNI
DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(MATHÉMATIQUES APPLIQUÉES)
AOÛT 2017

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

MODÈLE DE PRÉVISION DES TAUX DE CLICS DES ANNONCES TEXTUELLES SUR LES MOTEURS DE RECHERCHE

présenté par : SANNI Farooq
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de :

M. GAMACHE Michel, Ph. D., président
M. ADJENGUE Luc-Désiré, Ph. D., membre et directeur de recherche
M. LABIB Richard, Ph. D., membre

DÉDICACE

À ma famille.

REMERCIEMENTS

Je tiens tout particulièrement à remercier mon directeur de recherche M. Luc-Désiré Adjengue pour son support tout au long de ma maîtrise. En effet, son expertise et sa disponibilité ont été très précieuses dans la réalisation de ce projet.

Je remercie également mes parents pour leur amour, leur soutien moral et financier durant toutes ces années d'études. Merci à mes deux sœurs qui ont toujours fait preuve d'intérêt et d'encouragement dans mes travaux.

Je remercie aussi M. Olivier Gaudoin d'avoir accepté être mon tuteur à l'ENSIMAG dans le cadre de ma mobilité à Polytechnique Montréal. Enfin, mes remerciements vont à M. Michel Gamache et M. Richard Labib, respectivement président et membre de mon jury.

RÉSUMÉ

Le taux de clics est une métrique essentielle dans les campagnes publicitaires sur les moteurs de recherche. En effet, il impacte directement les deux acteurs principaux de la publicité en ligne que sont les moteurs de recherche d'un côté et les annonceurs de l'autre. D'une part le taux de clics est la principale variable utilisée par les moteurs de recherche dans leur algorithme d'affichage des annonces textuelles. Aussi leurs revenus sont intimement liés à l'ordre d'affichage des différentes annonces. De plus, proposer une publicité pertinente à un utilisateur améliore son expérience et l'incite à utiliser davantage le moteur de recherche. D'autre part, le taux de clics joue le rôle d'indice de qualité pour les annonceurs ; ces derniers ajustent les paramètres de leurs campagnes suivant les valeurs du taux de clics. Une bonne prédiction du taux de clics est alors très importante aussi bien pour les moteurs de recherche que pour les annonceurs.

Pour prédire le taux de clics, les moteurs de recherche disposent d'un historique riche et détaillé des réalisations des annonces textuelles. Les principales variables disponibles sont des variables catégoriques issues des informations sur les annonceurs, les utilisateurs ou encore des données géographiques. Dans ce mémoire, la régression logistique est appliquée deux fois pour prédire le taux de clics. Les données des campagnes publicitaires contiennent beaucoup d'observations à taux de clics nul complexifiant la modélisation. Ainsi, la première régression logistique permet d'écarter ces observations tandis que la seconde prédit le taux de clics des autres observations. Aussi des variables « inédites » sont utilisées dans ces deux régressions. En effet les variables *position moyenne*, *nombre d'impressions* et *coût* sont d'abord modélisées, puis elles sont utilisées comme variables explicatives dans le modèle logistique. Ces variables sont en réalité des variables de réponse tout comme le taux de clics. Ainsi nous proposons un modèle pour chacune de ces variables. La loi normale tronquée est ajustée à la position moyenne ; pour le nombre d'impressions et le coût, différents modèles sont explorés notamment les modèles linéaires généralisés (Poisson, Gamma, lognormal). Des modèles de type *hurdle* sont finalement retenus. Aussi, nous montrons qu'une hypothèse d'indépendance temporelle des observations, nécessaire à l'application de nos méthodes, est plausible malgré le phénomène de mesures répétées. Enfin les expériences menées sur des données réelles, montrent que cette modélisation en chaîne donne de bons résultats et peut encore être améliorée.

ABSTRACT

Click-through rate is an essential metric in advertising campaigns on search engines. As a matter of fact, it directly impacts the two main players of online advertising which are search engines and advertisers. On the one hand, the click-through rate is the main variable used by search engines in their algorithm for displaying text ads. Also their revenues are intimately linked to the order of display of the different ads. Additionally, offering relevant advertising to a user improves their experience and encourages them to make greater use of the search engine. On the other hand, the click-through rate plays the role of a quality score for advertisers who adjust their campaign settings based on click-through rate values. A good click-through rate prediction is very important for both search engines and advertisers.

To predict the click-through rate, search engines have a large amount of historical data on text ads. The main variables available are categorical variables derived from information about advertisers, users, or geographic data. In this paper, logistic regression is applied twice to predict the click-through rate. Campaign data contains many observations with zero clicks that make modeling more complex. The first logistic regression then discards these observations while the second predicts the click-through rate of the other observations. Also, new variables are used in these two regressions. Indeed the variables *mean position*, *number of impressions* and *cost* are first modeled then they are used as explanatory variables in the logistic model. These variables are actually response variables as the click-through rate. Thus, we propose a model for each of these variables. The truncated normal distribution is adjusted to the *mean position* ; for the *number of impressions* and the *cost*, different models are explored in particular some generalized linear models (Poisson, Gamma, lognormal). *Hurdle* models are finally retained. We also show that a hypothesis of temporal independence of observations, necessary for the application of our methods, is plausible despite the phenomenon of repeated measures. Finally, experiments carried out on real data show that this chain modeling gives good results and can be further improved.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

CPC	Cost Per Click
CTR	Click-Through Rate
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
MAE	Mean Abslute Error
LR	Likelihood Ratio
ACF	Auto-Correlation Function
PACF	Partial Auto-Correlation Function
ARMA	Autoregressive Moving Average
LASSO	Least Absolute Shrinkage and Selection Operator

LISTE DES ANNEXES

CHAPITRE 1 INTRODUCTION

L'accès à l'internet est aujourd'hui un acquis pour plus de 43% de la population mondiale selon les chiffres de l'année 2016 de l'Union Internationale des Télécommunications (UIT, 2016). Dans les pays dits développés, ce pourcentage atteint 81%. Ainsi au Canada par exemple, plus de quatre personnes sur cinq naviguent sur le web, effectuent des recherches et surtout sont susceptibles de voir des publicités. Internet, avec son large auditoire, apparaît alors comme un support incontournable pour effectuer la promotion d'un produit ou d'un service. C'est pourquoi l'investissement des entreprises dans les campagnes publicitaires en ligne pour promouvoir leurs produits ne cesse d'augmenter depuis plusieurs années. En effet la part de marché de la publicité en ligne ne cesse d'augmenter au détriment des supports traditionnels que sont la télévision, la radio et la presse (Bys, 2017). L'année 2016 a notamment marqué un tournant puisque pour la première fois, la publicité en ligne est passée devant la télévision en terme de part de marché (PwC, 2016).

Outre son large public, cet engouement pour la publicité en ligne s'explique aussi par la flexibilité qu'elle offre. D'abord, elle permet un meilleur ciblage des utilisateurs. De plus, plutôt que de s'adresser à un groupe comme c'est le cas des spots publicitaires, on peut ici s'adresser à chaque utilisateur en personnifiant la publicité grâce à l'analyse des *cookies* (petits fichiers stockés sur le terminal de l'utilisateur et contenant des informations personnelles) par exemple. Enfin, la publicité en ligne peut être plus facilement modifiée, ajustée.

On distingue essentiellement trois types de publicité en ligne : les annonces textuelles, les bannières et la vidéo. Les annonces textuelles sont associées au « réseau recherche » ; elles se présentent sous la forme d'un texte court (4 lignes maximum) et apparaissent au dessus des résultats d'une requête sur un moteur de recherche. Les bannières sont quant à elles associées au « réseau display » c'est-à-dire qu'elles sont diffusées sur différentes pages web à côté du contenu de ces pages. Enfin la vidéo est la forme de publicité qui est le plus en essor ; elles se retrouvent essentiellement sur les réseaux sociaux et sur les plateformes de vidéo telles que Youtube. C'est la première forme de publicité, les annonces textuelles, qui va nous intéresser dans ce présent mémoire.

Dans la suite de cette introduction, nous donnons la définition de quelques termes utiles à la compréhension puis nous décrivons le mécanisme de fonctionnement des annonces textuelles. Enfin nous explicitons notre problématique suivie de la description de notre projet de mémoire.

1.1 Définitions

Le monde de la publicité en ligne dispose d'un jargon dont une connaissance préalable est nécessaire. Aussi il est important d'expliquer clairement les termes qui sont utilisés tout au long de ce mémoire.

Notre travail est consécutif à ceux de Quinn (2011) et Assari (2014). Les définitions qui sont décrites ci-dessous sont également présentées dans leurs mémoires ; la section 1.2 de Quinn (2011) est particulièrement très exhaustive sur ces définitions. Ainsi la présentation ci-dessous est succincte mais suffisante à la compréhension. Le lecteur intéressé par plus de détails est invité à consulter la section 1.2 de Quinn (2011).
Nous divisons les définitions en une première partie portant sur la terminologie et une seconde présentant les variables d'intérêts dans le mécanisme des annonces textuelles.

1.1.1 Terminologie

Ici nous présentons quelques éléments de vocabulaire utilisés dans le contexte des campagnes de publicité sur les moteurs de recherche.

- Un **utilisateur** désigne un internaute, c'est-à-dire une personne qui utilise internet plus particulièrement un moteur de recherche dans notre cas.
- L'**annonceur** désigne l'entreprise ou la personne qui souhaite donner de la visibilité à un produit ou un service.
- Un **moteur de recherche** est « un outil de recherche qui référence automatiquement les pages web se trouvant sur le réseau Internet à l'aide d'un programme » (L'encyclopédie illustrée du marketing, 2017). L'utilisateur interroge le moteur de recherche en entrant des mots ; ce dernier lui retourne alors un ensemble de résultats jugés pertinents. Google est le plus important moteur de recherche et est utilisé par plus de 78% des utilisateurs (NetMarketShare, 2017). Derrière ce mastodonte, on peut citer Bing (8%), Baidu(8%), Yahoo(5%).
- Une **requête** est la recherche effectuée par l'utilisateur. Plus précisément c'est la suite de mots entrée dans la barre de recherche.
- Une **annonce textuelle** « constitue une forme de communication marketing que les annonceurs peuvent utiliser pour promouvoir leur produit ou service » (Google AdWords, 2017). Elle se retrouve uniquement dans les moteurs de recherche à la suite d'une requête effectuée par un utilisateur ; elle s'affiche avant les résultats dits organiques, c'est-à-dire ceux qui sont liés à la requête et qui ne sont pas de la publicité. Elle se présente sous la forme d'un texte décrivant le produit et propose un lien vers le

site de l'annonceur. La figure 1.1 montre quatre annonces textuelles obtenus suite à la requête « assurance habitation » sur le moteur de recherche Google.

- Un **mot clé** est un ensemble de mots que l'annonceur fournit au moteur de recherche et qui décrit son produit. Ainsi lorsqu'un utilisateur effectue une recherche dont les termes sont similaires à ceux du mot clé, l'annonce associée est alors susceptible de s'afficher. Le degré de similitude entre la requête et le mot clé est défini par le type de correspondance du mot clé. On distingue entre autres les options « requête large », « mot clé exact », etc. Ce paramètre nous intéresse peu car nous ne travaillons qu'avec la correspondance « requête large ». Dans le cas de la « requête large », les fautes d'orthographe ainsi que les synonymes entre le mot clé et les requêtes sont acceptés. Par exemple la requête « manteau dames » affichera l'annonce associée au mot clé « manteaux femmes ».
- Une **campagne** est un regroupement de mots clés d'un annonceur qui partagent des paramètres tels que le budget et le ciblage géographique.

1.1.2 Mécanisme de fonctionnement des annonces textuelles

Avant de décrire le mécanisme de fonctionnement des annonces textuelles, nous définissons d'abord les éléments relatifs à celui-ci.

- Une **impression** correspond à l'affichage d'une annonce textuelle. Cette impression provient de la correspondance entre un mot clé en particulier et la requête d'un utilisateur. On distingue alors le nombre d'impressions d'une annonce de celui d'un mot clé. Le nombre d'impressions d'une annonce textuelle est le nombre de fois où elle est apparue tandis que le nombre d'impressions d'un mot clé est le nombre de fois que ce mot clé a généré l'impression d'une annonce. Le nombre d'impressions d'une annonce est partagé entre ses mots clés. Dans la suite, le nombre d'impressions désignera implicitement le nombre d'impressions d'un mot clé.
- On parle de **clic** lorsqu'un utilisateur clique sur une annonce textuelle. Le mot clé ayant conduit à ce clic voit son nombre de clics incrémenté d'une unité. La distinction précédente faite pour les nombres d'impressions s'applique également au nombre de clics.
- Le **taux de clics** ou CTR (Click-through Rate) est le rapport du nombre de clics sur celui d'impressions. Il indique la proportion d'utilisateurs qui voit une annonce et clique dessus. Un CTR élevé est un bon indicateur de la qualité et de la pertinence d'un mot clé.
- La **position** est la place occupée par l'annonce au moment de son affichage. Il s'agit donc de valeurs entières. La position la plus élevée est 1 indiquant que l'annonce appa-

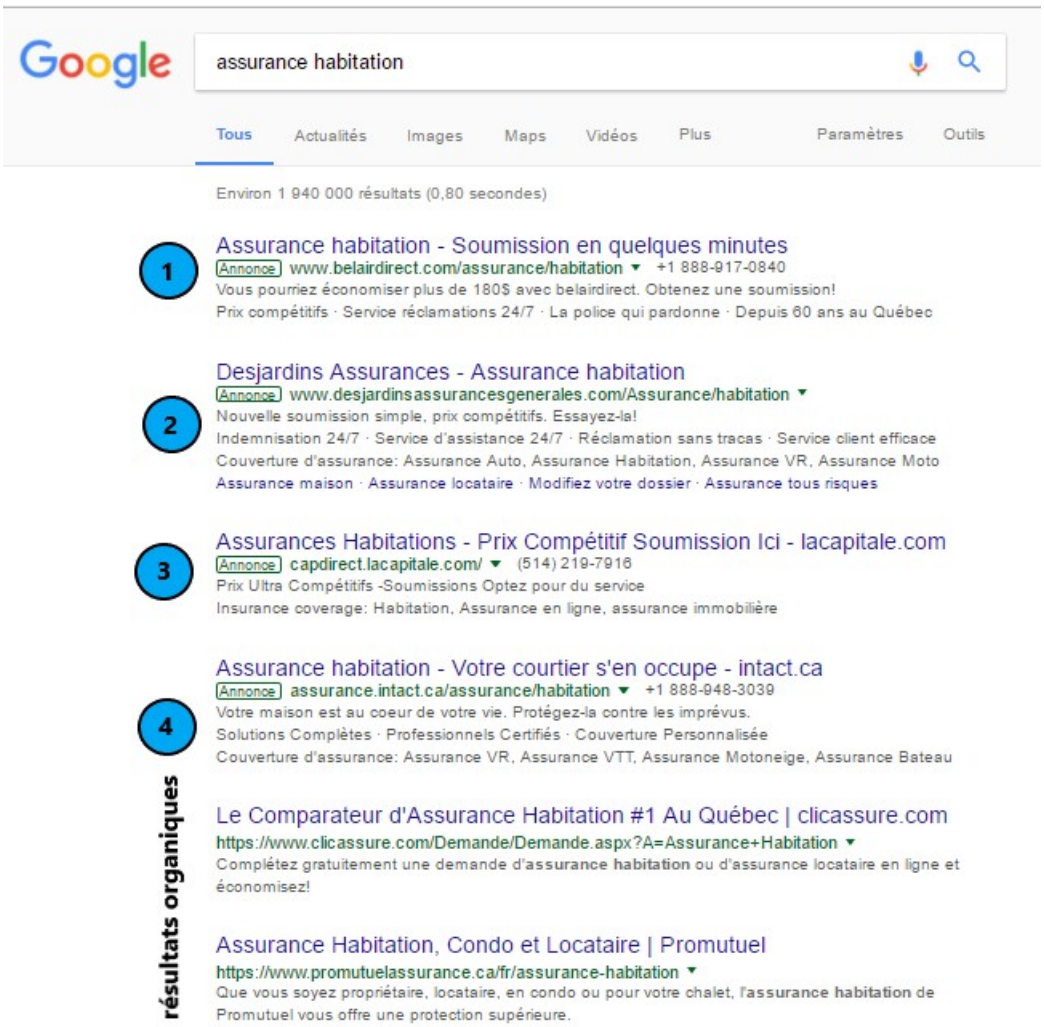

Figure 1.1 Résultats d'une requête dans le moteur de recherche Google.

raît tout en haut de la première page de résultats. Elle peut être très grande (supérieure à 20) puisque des annonces sont également affichées sur les autres pages de résultats. On retrouve entre 1 et 8 annonces sur la première page. De plus, la plupart des utilisateurs ne vont pas au delà de la deuxième page donnant donc des valeurs de positions entre 1 et 10. Sur la figure 1.1, nous avons marqué en face de chaque annonce sa position pour illustrer celle-ci.

Pour chaque mot clé, les moteurs de recherche calculent la moyenne des différentes positions occupées au cours d'une journée. On se retrouve alors avec une position moyenne qui n'est pas toujours entière dans les données disponibles.

- Dans ce mémoire, le **coût** désigne la somme payée pour un clic sur une annonce (CPC). Il existe la tarification au clic et la tarification au millier d'impressions. C'est la première

que nous étudions. Une quantité importante est le coût par clic maximal (max CPC) : il s'agit du montant maximal qu'un annonceur est prêt à payer pour obtenir un clic.

- Lorsqu'un utilisateur clique sur une annonce, il est redirigé vers un site web où il est invité à effectuer une action telle qu'un achat, une souscription à un abonnement, un visionnage, etc. On parle de **conversion** lorsque cette action est complétée.

Soulignons que les mesures de ces quantités sont agrégées sur une journée. En effet les moteurs de recherche fournissent aux annonceurs des statistiques quotidiennes : un nombre d'impressions quotidien, un coût quotidien, etc.
Une fois les termes importants définis, une brève description du fonctionnement des annonces textuelles est décrite ci-dessous.

Une requête dans un moteur de recherche génère en plus des résultats, des annonces publicitaires. L'apparition d'une annonce est déterminée à la suite d'un processus d'enchères effectué par un algorithme du moteur de recherche. La valeur d'enchère d'un mot clé est le produit du max CPC, fixé par l'annonceur, et d'un indice de qualité calculé par le moteur de recherche. Les annonces sont alors affichées par ordre décroissant de leurs valeurs d'enchères. Deux composantes interviennent donc dans l'affichage d'une annonce : d'abord le max CPC contrôlé par l'annonceur qui peut décider de l'augmenter ou de le diminuer, puis l'indice de qualité. Le calcul de cet indice diffère d'un moteur de recherche à l'autre ; il est précieusement gardé par les moteurs de recherche. On sait néanmoins qu'il dépend de la pertinence des mots clés associés à l'annonce, de l'historique du taux de clics de l'annonce, de considérations géographiques et d'autres facteurs.
Généralement, les premières positions sont celles qui génèrent le plus de clics (Agichtein et al., 2006; Joachims et al., 2005; Craswell et al., 2008). Ainsi il y a une concurrence naturelle qui apparaît entre les annonceurs afin que leurs annonces occupent ces places. Mais on voit déjà qu'à cause de l'indice de qualité, il ne suffit pas de fixer un max CPC élevé pour atteindre les premières places. Aussi il faut qu'une campagne publicitaire soit rentable c'est-à-dire que les revenus attendus soient supérieurs aux montants dépensés dans la publicité.

1.2 Problématique et présentation du projet

La qualité d'une annonce est jugée par son taux de conversion c'est-à-dire le rapport du nombre de conversions sur celui d'impressions. Néanmoins ce critère s'avérant compliqué à mesurer car propre à chaque entreprise, on lui préfère le taux de clics. Aussi on suppose implicitement une corrélation positive entre ces deux taux c'est-à-dire que le taux de conversion se comporte de manière similaire au taux de clics (un taux de clics élevé traduit un taux de conversion élevé). Ainsi une entreprise qui lance sa campagne de publicité souhaite maximiser

le taux de clics de ses mots clés sous une certaine contrainte budgétaire.

Le problème à résoudre ici est un problème d'optimisation sous-contraintes. Beaucoup d'études se sont donc penchées sur sa résolution. Nous en parlons plus en détails dans la revue de la littérature. Néanmoins la présence de beaucoup d'inconnues notamment l'algorithme d'affichage des annonces des moteurs de recherche rend le problème assez complexe.

Notre hypothèse est qu'une étude statistique approfondie des données apporte des informations supplémentaires qui permettent d'améliorer la résolution de ce problème. Cette composante statistique du problème a été moins traitée dans la littérature. Dans le cadre de ce projet, nous proposons une approche basée sur une double régression logistique pour prédire le taux de clics d'un mot clé à partir uniquement d'un historique des réalisations de ses positions moyennes, ses nombres d'impressions, ses nombres de clics et ses coûts. La plupart des études statistiques menées sur les annonces textuelles sont plutôt axées sur les utilisateurs c'est-à-dire qu'on cherche à afficher la « meilleure annonce » à chaque utilisateur. Par meilleure annonce, on entend l'annonce sur laquelle l'utilisateur est le plus susceptible de cliquer. Le modèle proposé ici ne fait pas intervenir les utilisateurs : ce qui est très utile car les informations sur ces derniers ne sont pas toujours disponibles. Une modélisation des variables explicatives utilisées dans notre modèle à savoir la position, les impressions et le coût est nécessaire car tout comme le taux de clics, elles ne sont pas connues à l'avance. En effet, tout comme au jour j le taux de clics d'un mot clé au jour suivant $j+1$ est inconnu, la position, le nombre d'impressions et le coût sont également inconnus au jour $j+1$. Ainsi un modèle est ajusté sur chacune de ces trois variables : une loi normale tronquée est ajustée aux positions ; un modèle de type *hurdle* logistique et binomial négatif est choisi parmi une sélection de modèles pour estimer les impressions. Enfin pour le coût, le modèle *hurdle* logistique et Gamma est retenu.

Dans la suite de ce mémoire, une revue de la littérature est d'abord présentée au chapitre 2. En plus de présenter l'état de l'art, elle permet de mettre en exergue l'originalité de notre recherche. Ensuite le chapitre 3 est consacré à la structure du modèle de prévision. Une brève mais complète présentation de la régression logistique permet d'introduire et de détailler le modèle de prévision des taux de clics. Puis nous décrivons les jeux de données dont nous disposons et sur lesquelles nos méthodes seront appliquées. Les modèles intermédiaires utilisées pour estimer les variables explicatives sont décrits dans le chapitre 4. Enfin les résultats de l'application de notre méthode sur les jeux de données sont discutés au chapitre 5 ; ils seront suivis d'une conclusion au chapitre 6.

CHAPITRE 2 REVUE DE LITTÉRATURE

L'essor de la publicité en ligne s'est accompagné d'une augmentation des activités de recherche sur le sujet. Il s'agit d'un domaine relativement récent puisqu'il apparaît avec le développement d'Internet au début des années 2000. Aujourd'hui on compte un grand nombre d'articles et de publications sur la publicité en ligne et les annonces textuelles notamment.

D'abord il faut noter qu'il existe une diversité de sujets d'intérêts dans le domaine des annonces textuelles. Nous avons par exemple l'optimisation des enchères qui consiste à trouver la valeur d'enchère optimale à attribuer à un mot clé afin d'atteindre une position souhaitée. On peut citer entre autres les travaux de Borgs et al. (2007), de Zhou et al. (2008). Aussi, maximiser le nombre de clics total d'une campagne sous une contrainte budgétaire est également beaucoup étudié dans la littérature. Archak et al. (2010) proposent une méthode basée sur des processus de décision de Markov ; DasGupta et Muthukrishnan (2013) quant à eux utilisent des méthodes d'optimisation stochastique.

Quinn (2011) d'abord et Assari (2014) ensuite ont travaillé sur ces deux problématiques. En effet dans ces deux mémoires de maîtrise (réalisés à Polytechnique Montréal), ils s'intéressent à l'optimisation des campagnes publicitaires, c'est-à-dire maximiser le rendement des annonceurs. Assari (2014) introduit des techniques de fouille de données afin d'améliorer le modèle proposé par Quinn (2011). Leur problématique est totalement différente de la nôtre puisque c'est la prédiction du taux de clics qui nous intéresse. Dans la littérature récente, la prédiction des taux de clics est le sujet principalement traité. En effet une connaissance du taux de clics facilite la résolution des problématiques précédentes.

Le taux de clics est une quantité très importante à la fois pour les annonceurs et les moteurs de recherche. Il sert d'indice de qualité pour les annonceurs ; il permet à ces derniers d'ajuster leurs campagnes, modifier les valeurs d'enchère de certains mots clés ou encore de retirer des mots clés très peu pertinents. Quant aux moteurs de recherche, il utilise le taux de clics dans leur algorithme d'affichage des annonces. C'est une des variables qui permet de déterminer l'ordre d'apparition des annonces. C'est ainsi que les principaux articles publiés sur la prédiction des taux de clics proviennent des compagnies propriétaires des moteurs de recherche. Toutefois en raison de la compétition entre ces différentes compagnies, celles-ci ne publient qu'une petite partie de leur travaux.

La régression logistique est l'un des principaux modèles utilisés pour prédire le taux de clics. Richardson et al. (2007), Chapelle et al. (2015) appliquent la régression logistique. Chapelle et al. (2015) utilisent uniquement des variables qualitatives ; ils incluent les informations sur

les annonces, les annonceurs, les utilisateurs et également le temps. Ils disposent également des données pour chaque impression ; aucune agrégation des données n'est opérée. À chaque impression, on sait si l'utilisateur clique ou non sur l'annonce. C'est la différence fondamentale entre les recherches menées par les moteurs de recherche et celles menées par des chercheurs indépendants. Les premiers disposent de données beaucoup plus riches.

Chez Google également on utilise la régression logistique puisque McMahan et al. (2013) proposent l'algorithme FTRL-proximal (*Follow The Proximally Regularized Leader*) qui permet d'obtenir un modèle « creux ». En effet, l'ajustement d'un modèle logistique fournit un vecteur de paramètres pleins, c'est-à-dire que peu de paramètres sont nuls. L'algorithme proposé permet d'avoir plus de paramètres nuls sans toutefois trop dégrader le modèle ; il améliore également la convergence. McMahan et al. (2013) suggèrent également des techniques pour améliorer les implémentations notamment une économie de mémoire. En effet les moteurs de recherche ont des milliers voire des millions d'annonceurs, donc beaucoup de données à manipuler. Ainsi les modèles creux sont très intéressants.

Outre la régression logistique, les moteurs de recherche emploient également d'autres méthodes pour prédire le taux de clics. Graepel et al. (2010) utilisent la régression probit qui est aussi un modèle binomial comme le modèle logistique. Ils utilisent essentiellement des données catégoriques. Zhu et al. (2010) proposent une méthode basée sur les réseaux bayésiens pour notamment prédire le taux de clics des mots clés très peu utilisés. Chez Yandex, un moteur de recherche russe, on utilise plutôt des arbres « boostés » (*boosted trees*) (Trofimov et al., 2012) qui sont une modification des machines à gradient « boosté ». Un réseau de neurones est ensuite introduit pour traiter les variables catégoriques : Baqapuri et Trofimov (2014) utilisent un réseau de neurones qui prend les variables catégoriques en entrée et calcule une première estimation du taux de clics. Puis le modèle des arbres « boostés » utilise cette estimation et les autres variables pour finalement prédire le taux de clics.

Jusqu'ici nous avons présenté différents modèles disponibles dans la littérature. Cependant il est presque impossible de les comparer car d'une part les données utilisées ne sont pas disponibles. D'autre part chaque moteur de recherche dispose de son propre algorithme d'affichage des annonces. Il est alors très probable que le modèle de Yandex échoue pour des données issues du moteur de recherche de Google par exemple.

Parallèlement aux travaux des moteurs de recherches, des recherches indépendantes sont également menées notamment par les entreprises de gestion de campagnes publicitaires. Comme nous le soulignions plus haut, ici les données disponibles sont beaucoup moins riches puisqu'elles sont fournies par les moteurs de recherches sous forme agrégées. Par exemple, pour un mot clé on dispose non pas des informations pour chaque impression mais plutôt une

moyenne de ces informations sur une certaine période donnée (une journée généralement). La qualité de ces données rend la tâche de prédiction ici beaucoup plus compliquée. C'est en partie pour cela que ce sujet a longtemps été contourné en formulant les problématiques autrement telles que la maximisation du nombre de clics. Ici aussi les plateformes de gestion de campagnes ne sont pas très enclines à partager leurs travaux. Toutefois on retrouve quelques articles très intéressants.

Kumar et al. (2015) utilisent la régression logistique et surtout introduisent la position comme une variable explicative. Lee et al. (2012) proposent d'utiliser un modèle pour traiter l'emboitement des variables catégoriques. Même s'ils s'intéressent aux conversions, leur modèle reste transposable aux clics. L'émergence de l'apprentissage profond a conduit Jiang (2016) à proposer un modèle combinant un *Deep belief network* et un modèle logistique. Tout comme Baqapuri et Trofimov (2014), le réseau de neurones permet ici de tirer de l'information des variables catégoriques. Au vu des résultats des réseaux de neurones notamment les réseaux profonds sur différentes tâches de l'apprentissage automatique, nous pensons que de plus en plus de modèles basés sur les réseaux de neurones seront proposés pour prédire les taux de clics. Ces modèles sont beaucoup plus complexes mais difficiles à interpréter. Enfin on peut citer le travail de Lee et al. (2016) qui proposent des modèles linéaires avec noyau comme alternative au modèle logistique.

En somme le modèle logistique est un modèle de référence pour prédire les taux de clics. D'un côté, nous avons les moteurs de recherche qui disposent de données détaillées et de l'autre des chercheurs indépendants avec des données moins fournies. Nous nous situons dans le deuxième groupe. La plupart des méthodes disponibles dans la littérature utilisent principalement des variables catégoriques. Or ne disposant pas des données sur ces variables au départ de notre projet, nous avons donc bâti un modèle qui n'utilise quasiment pas ces dernières même si par la suite ces données furent disponibles. Nous utilisons également le modèle logistique. Néanmoins nous proposons des modèles pour les variables intermédiaires que sont la position moyenne du mot clé, son nombre d'impressions ou encore le coût d'un clic. Cette méthode d'estimation du taux de clics est assez inédite.

Dans le chapitre suivant, nous faisons une introduction détaillée au modèle logistique puis nous présentons notre modèle de prédiction des taux de clics.

CHAPITRE 3 STRUCTURE DU MODÈLE DE PRÉVISION

Nous proposons une nouvelle méthode pour prédire le taux de clics des mots clés. Cette méthode repose principalement sur le modèle logistique qui est un modèle de régression non linéaire. Nous appliquons la régression logistique deux fois : une première fois comme un classificateur multiclasse et une seconde fois comme un modèle de régression. Afin de présenter notre méthode, une compréhension du modèle de régression logistique est nécessaire. Ainsi dans la suite, nous faisons d'abord une introduction de la régression logistique ; nous y présentons les principaux éléments du modèle. Ensuite nous explicitons en détails notre modèle de prévision des taux de clics. Puis nous montrons la validité des hypothèses faites par la régression logistique dans notre contexte. Enfin nous présentons les données sur lesquelles notre modèle sera testé.

3.1 La régression logistique

La régression logistique est un modèle de régression qui permet d'établir le lien entre une réponse catégorique et des variables indépendantes. La réponse peut admettre deux ou plusieurs catégories. Nous nous intéressons d'abord au cas où la variable dépendante admet deux catégories : par exemple un étudiant réussit ou échoue à un examen, un patient survit ou meurt etc. Suivant un certain nombre de caractéristiques, on veut déterminer la probabilité qu'une observation appartienne à une catégorie. La régression logistique est utilisée dans des domaines très variées telles que la médecine, la finance. Par exemple Hosmer et al. (2013) cherchent à déterminer la présence ou l'absence de maladie coronarienne chez des patients en fonction de leur âge ; James et al. (2013) quant à eux appliquent la régression logistique pour savoir si un client détenteur d'une carte de crédit risque d'être à découvert ou pas.

3.1.1 Présentation du modèle

Soit Y une variable de réponse binaire. On notera par "0" et "1" les deux valeurs possibles que peut prendre Y ; on parle généralement de classes pour désigner ces valeurs. Notons $X = (X_1, X_2, \ldots, X_k)^T$ le vecteur des variables explicatives ou variables indépendantes. On pose également $p(\mathbf{x}) = P(Y = 1|X = \mathbf{x})$ la probabilité que Y appartienne à la classe "1" étant donné l'entrée $\mathbf{x}$. La variable Y étant donné $\mathbf{x}$ suit alors une loi de Bernoulli de paramètre $p(\mathbf{x})$; on note $Y|\mathbf{x} \sim \mathcal{B}(p(\mathbf{x}))$. Ainsi l'observation caractérisée par le vecteur des entrées $\mathbf{x}_0$ sera classée dans la classe "1" si $p(\mathbf{x}_0) > 0,5$ et dans la classe "0" sinon.

Le modèle de régression logistique est un modèle linéaire généralisé basée sur la transformation *logit* de la moyenne $\mathbb{E}(Y|X) = p(X)$. Il s'écrit :

$$\begin{aligned} \log\left(\frac{p(X)}{1-p(X)}\right) &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k \\ &= X^T\beta, \end{aligned} \tag{3.1}$$

où $\beta_0, \beta_1, \ldots, \beta_k$ sont des paramètres réels. Dans l'équation 3.1, $\beta = (\beta_0, \beta_1, \ldots, \beta_k)^T$ est le vecteur des paramètres et le vecteur X est modifié en ajoutant 1 au début tel que $X = (1, X_1, X_2, \ldots, X_k)^T$. Notons qu'une variable explicative X_j peut être une transformation non linéaire des autres variables. Par exemple $X_2 = X_1^2$ ou $X_3 = X_1\,X_2$. Lorsqu'une variable est le produit d'autres variables, on parle d'effet d'interaction ou simplement interaction : $X_3 = X_1\,X_2$ est l'interaction entre les variables X_1 et X_2.

Le rapport $\frac{p(X)}{1-p(X)}$ est appelé rapport des côtes ou *odds ratio*. La fonction *logit* permet le passage de l'intervalle $]0,1[$ à $\mathbb{R}$ afin de créer un lien entre une fonction de la moyenne $p(X)$ et les variables explicatives $X_1, X_2, \ldots, X_k$. La fonction logistique ou sigmoïde permet de retrouver la probabilité $p(X)$. En effet de l'équation 3.1, on obtient :

$$p(X) = \frac{1}{1+\exp\left(-X^T\beta\right)}. \tag{3.2}$$

Comme le montre la figure 3.1, la sigmoïde ramène les valeurs de $\mathbb{R}$ dans l'intervalle ouvert $]0,1[$.

3.1.2 Estimation des paramètres

Afin de pouvoir utiliser $p(\mathbf{x})$ donnée par l'équation 3.2, il est nécessaire que les paramètres $\beta_0, \beta_1, \ldots, \beta_k$ du modèle 3.1 soient estimés puisqu'en général ils sont inconnus. L'estimation de ces paramètres se fait par la méthode de vraisemblance maximale. La méthode de vraisemblance maximale consiste à déterminer la valeur $\hat{\beta}$ du paramètre $\beta = (\beta_0, \beta_1, \ldots, \beta_k)^T$ qui maximise la fonction de vraisemblance.

Considérons n observations indépendantes de la forme $\{(\mathbf{x}_i, y_i), i = 1, 2, \ldots, n\}$. On suppose qu'on dispose de k variables explicatives : chaque $\mathbf{x}_i, 1 \leq i \leq n$ est un vecteur de taille k c'est-à-dire que $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ik})^T$. Aussi les y_i sont codés tels que pour tout $i = 1, \ldots, n$, $y_i \in \{0, 1\}$. Pour $i = 1, \ldots, n$, $Y_i|\mathbf{x}_i \sim \mathcal{B}(p_i)$, où $p_i = P(Y_i = 1|\mathbf{x}_i)$ représente la probabilité que Y_i soit égal à 1. La fonction de masse de l'échantillon s'écrit :

$$p(y_i; \beta) = p_i^{y_i}(1-p_i)^{1-y_i}, \quad y_i \in \{0,1\},\ i = 1, \ldots, n. \tag{3.3}$$

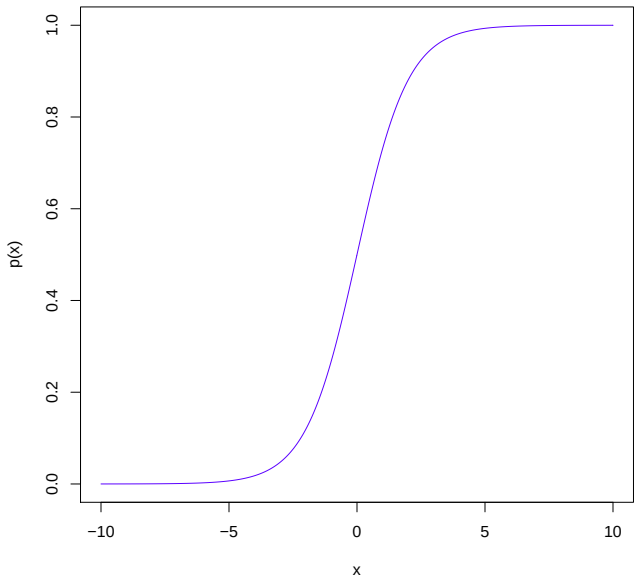

Figure 3.1 Graphe de la fonction logistique ou sigmoïde $p(\mathbf{x}) = \frac{1}{1+\exp(-x)}$, $x \in [-10, 10]$.

Lorsque l'hypothèse d'indépendance des observations est vérifiée, la fonction de vraisemblance s'obtient en faisant le produit des fonctions de masse des n observations comme suit :

$$\mathcal{L}(\beta; y_1, y_2, \ldots, y_n) = \prod_{i=1}^{n} p(y_i; \beta) = \prod_{i=1}^{n} p_i^{y_i}(1 - p_i)^{1-y_i}. \tag{3.4}$$

On maximise généralement le logarithme de la vraisemblance que nous désignons par log vraisemblance dans la suite de ce mémoire. De l'équation 3.4, la log vraisemblance est :

$$\ell(\beta; y_1, y_2, \ldots, y_n) = \sum_{i=1}^{n} \left[y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \right]. \tag{3.5}$$

Ainsi $\hat{\beta} = \arg\max_{\beta} \ell(\beta; y_1, y_2, \ldots, y_n)$.

La log vraisemblance est une fonction concave ; elle admet donc un unique maximum. Celui-ci s'obtient alors en annulant la dérivée de la log vraisemblance par rapport à β. En remplaçant p_i par son expression donnée par l'équation 3.2 dans la log vraisemblance (équation 3.5) et en dérivant par rapport à β on obtient (Friedman et al., 2001) :

$$\frac{\partial \ell(\beta; y_1, y_2, \ldots, y_n)}{\partial \beta} = \sum_{i=1}^{n} \mathbf{x}_i (y_i - p_i) = \mathbf{0}. \tag{3.6}$$

Cette équation n'admet pas de solution analytique. On utilise alors des méthodes d'optimisation numérique notamment l'algorithme de Newton-Raphson. Il s'agit d'une procédure itérative du gradient qui fait intervenir la matrice hessienne $\frac{\partial \ell(\beta)}{\partial \beta \partial \beta^T}$, où $\ell(\beta)$ représente $\ell(\beta; y_1, y_2, \ldots, y_n)$. L'algorithme se présente en deux étapes comme suit :

1. Initialisation $\beta = \beta^{(0)}$
2. Tant que le critère de convergence n'est pas vérifié (norme du vecteur gradient inférieur à un seuil ϵ par exemple) :
   - Évaluer le vecteur gradient $\frac{\partial \ell(\beta)}{\partial \beta}$ et la matrice hessienne $\frac{\partial \ell}{\partial \beta \partial \beta^T}$ en $\beta^{(k)}$.
   - Mettre à jour β : $\quad \beta^{(k+1)} = \beta^{(k)} - \left(\frac{\partial \ell(\beta)}{\partial \beta \partial \beta^T}\right)^{-1} \frac{\partial \ell(\beta)}{\partial \beta}$.

En posant $\mathbf{y} = (y_1, y_2, \ldots, y_n)^T \in \mathbb{R}^n$, $\mathbf{p} = (p_1, p_2, \ldots, p_n)^T \in \mathbb{R}^n$, C une matrice carrée d'ordre n telle que $c_{ii} = p_i(1 - p_i)$ et $c_{ij} = 0$, $i \neq j$ et

$$\mathbf{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1k} \\ 1 & x_{21} & x_{22} & \ldots & x_{2k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \ldots & x_{nk} \end{pmatrix},$$

on peut montrer que le vecteur gradient et la matrice hessienne sont alors (Hosmer et al., 2013) :

$$\frac{\partial \ell(\beta)}{\partial \beta} = \mathbf{X}^T(\mathbf{y} - \mathbf{p}) \quad ; \quad \frac{\partial \ell(\beta)}{\partial \beta \partial \beta^T} = -\mathbf{X}^T C \mathbf{X}. \tag{3.7}$$

Notons que le vecteur $\mathbf{p}$ est fonction de β, c'est-à-dire que $\mathbf{p} = \mathbf{p}(\beta^{(k)})$ permettant ainsi la mise à jour du vecteur gradient et de la matrice hessienne à chaque itération. L'équation de mise à jour du paramètre β devient alors :

$$\beta^{(k+1)} = \beta^{(k)} + (\mathbf{X}^T C \mathbf{X})^{-1} \mathbf{X}^T (\mathbf{y} - \mathbf{p}).$$

Plusieurs logiciels permettent d'obtenir l'estimation du paramètre β. Suivant l'implémentation de l'algorithme ou encore le choix de la méthode d'optimisation utilisée les valeurs peuvent différer légèrement. Nous utilisons le logiciel R pour estimer les paramètres.

En tant qu'estimateur de maximum de vraisemblance, $\hat{\beta}$ possède quelques propriétés intéressantes. En effet c'est un estimateur asymptotiquement sans biais et de variance minimale ; il est également convergent et suit asymptotiquement une loi normale. Cette dernière propriété est très utile dans la vérification de l'ajustement du modèle et l'inférence statistique.

3.1.3 Vérification du modèle

Afin de vérifier la performance d'une méthode de classification, un certain nombre de critères peuvent être utilisés notamment le taux d'erreur ou de mauvaise classification et la matrice de confusion.

Taux d'erreur ou de mauvaise classification

La régression logistique est généralement utilisée dans un contexte de classification, c'est-à-dire que pour une observation donnée, on cherche la classe à laquelle elle appartient. On peut mesurer la justesse (ou la précision) d'une méthode de classification en calculant son taux d'erreur ou de mauvaise classification. Le taux d'erreur se définit comme :

$$\text{taux d'erreur} = \frac{\text{nombre d'observations mal classées}}{\text{nombre total d'observations}}. \tag{3.8}$$

Un taux d'erreur bas signifie que le modèle est globalement bon dans la tâche de classification. Pour une évaluation plus précise d'un modèle de prévision donné, une méthode de calcul du taux d'erreur (équation 3.8) généralement utilisée dans les techniques d'apprentissage statistique consiste à diviser le jeu de données disponible en deux ensembles : un ensemble d'apprentissage et un ensemble de test. Le modèle est alors ajusté avec les données de l'ensemble d'apprentissage et le taux d'erreur est calculé avec celles de l'ensemble de test. Pour diminuer la sensibilité du calcul au choix de ces deux ensembles, des méthodes de validation croisée telles que le *Leave-One-Out* ou le *K-fold cross validation* (James et al., 2013) sont généralement utilisées.
Une analyse plus fine du classificateur se fait avec la matrice de confusion.

Matrice de confusion

La matrice de confusion est un tableau qui confronte les vraies classes et celles prédites ; elle permet de voir plus en détails les erreurs faites par le classificateur (modèle de classification). Le tableau 3.1 illustre la structure d'une matrice de confusion. Ici les classes sont notées "+" et "-". Une observation de la classe "+" sera dite positive et celle de la classe "-" négative. Il s'agit simplement de conventions pour introduire les quantités que nous présentons plus bas.

Tableau 3.1 Matrice de confusion.

		prédites	
		+	-
réelles	+	a	b
	-	c	d

Dans le tableau 3.1, **a** est le nombre de vrais positifs c'est-à-dire les observations classées positives et qui sont effectivement positives ; **c** est le nombre de faux positifs, les observations classées positives mais qui sont en réalité négatives. Dans le même ordre d'idée, **b** et **d** représentent respectivement le nombre de faux négatifs et de vrais négatifs. Ces quantités permettent de calculer quelques critères de performance (Rakotomalala, 2011).

- Le taux de vrais positifs est appelé sensibilité ou rappel ; il mesure la capacité du modèle à trouver les positifs. On a :
$$\text{sensibilité} = \frac{a}{a+b}. \tag{3.9}$$
- La spécificité quant à elle mesure la capacité du modèle à retrouver les négatifs ; c'est aussi le taux de vrais négatifs. Il est défini par
$$\text{spécificité} = \frac{d}{c+d}. \tag{3.10}$$
- La précision est la proportion de vrais positifs parmi les observations classées positives. On a
$$\text{précision} = \frac{a}{a+c}. \tag{3.11}$$
C'est une estimation de la probabilité qu'une observation classée positive le soit effectivement.

Pour les observations positives, puisqu'on a deux métriques à considérer en même temps, la F-mesure permet de les combiner en prenant leur moyenne harmonique. Aussi, elle permet d'accorder plus d'importance à l'une ou à l'autre grâce à un paramètre α. Son expression est (Sasaki et al., 2007) :
$$F_\alpha = \frac{(1+\alpha^2)(\text{précision} \times \text{sensibilité})}{\alpha^2 \times \text{précision} + \text{sensibilité}}. \tag{3.12}$$
Par exemple $\alpha = 1$ signifie que la même importance est accordée au rappel et à la précision. Une valeur de α supérieure à 1 ($\alpha = 2$ par exemple) donne plus d'importance au rappel (le rappel est deux fois plus important par rapport à la précision) tandis que pour α inférieur à 1 c'est la précision qui est avantagée. La F-mesure nous permettra au chapitre 5 de choisir la valeur optimale d'un hyperparamètre important.

Il existe d'autres mesures de performance telles que la courbe ROC qui représente la sensibilité (équation 3.9) en fonction de la spécificité (équation 3.10) . Ces différents critères de performance ne sont cependant pas suffisants pour mesurer la qualité de l'ajustement. Une analyse statistique est également nécessaire pour s'assurer notamment que le modèle est globalement significatif. L'analyse comprend : le test global, les tests individuels sur les coefficients et les pseudo-coefficients de détermination.

Test global

Le test global permet de vérifier l'utilité du modèle : au moins une des variables explicatives expliquent-elles la réponse Y ? Pour répondre à cette question, les hypothèses suivantes sont alors confrontées :

$$\begin{aligned} H_0 &: \beta_1 = \beta_2 = \ldots = \beta_k = 0 \\ \text{contre } H_1 &: \text{au moins un des } \beta_j \neq 0,\ j = 1, \ldots, p. \end{aligned}$$

L'acceptation de l'hypothèse nulle H_0 signifie qu'aucune des variables n'est utile dans la prédiction de la variable Y selon le modèle. Le test utilisé dans ce contexte est celui du rapport des vraisemblances. La statistique du test est (Hosmer et al., 2013) :

$$LR = -2 \times \log\left(\frac{\mathcal{L}(M_k)}{\mathcal{L}(M_0)}\right), \tag{3.13}$$

où $\mathcal{L}(M_k)$ est la vraisemblance du modèle avec les k variables et $\mathcal{L}(M_0)$ celle du modèle nul c'est-à-dire le modèle avec uniquement la constante β_0. Sous H_0, la statistique LR suit une loi de Khi-deux à k degrés de liberté. La règle de décision au seuil critique α (à ne pas confondre avec le α de la F-mesure) consiste alors à rejeter H_0 si $LR > \chi_{\alpha;k}$ où $\chi_{\alpha;k}$ est tel que $P(\chi_k > \chi_{\alpha;k}) = \alpha$. On peut alternativement utiliser la p-valeur$=P(\chi_k^2 > LR)$: on rejette H_0 si la p-valeur est inférieure à α.

Dans un modèle de régression logistique, une quantité d'intérêt est la déviance. C'est cette quantité qui est en général fournie par les logiciels statistiques. Pour un modèle M, elle se définit comme suit :

$$\begin{aligned} D =& D(M) = -2 \times \log\left(\frac{\text{vraisemblance du modèle}}{\text{vraisemblance du modèle saturé}}\right) \\ D =& -2 \times \log \text{ vraisemblance du modèle} + 2 \times \log \text{ vraisemblance du modèle saturé} \\ D =& D_M - (-2 \times \log \text{ vraisemblance du modèle saturé}), \end{aligned} \tag{3.14}$$

où $D_M = -2 \times$ log vraisemblance du modèle est appelée déviance résiduelle. Le modèle saturé est celui qui prédit exactement les données c'est-à-dire que $\hat{p_i} = p_i$ pour $i = 1, \ldots, n$. Ici la vraisemblance du modèle saturé est égale à 1 ; d'où la nullité de la log vraisemblance. Dans notre cas, la déviance est alors égale à la déviance résiduelle, c'est-à-dire $D = D_M$. La statistique 3.13 se réécrit en fonction de la déviance comme :

$$LR = D(M_0) - D(M_k) = D_{M_0} - D_{M_k}.$$

La déviance joue le rôle des sommes de carrés dans un modèle linéaire. La déviance $D(M_0)$ du modèle nul (modèle avec uniquement la constante β_0) s'identifie à la somme des carrés totale dans le modèle linéaire tandis que la déviance résiduelle D_M s'apparente à la somme des carrés des résidus que l'on cherche à minimiser.

Tests individuels sur les coefficients

On s'intéresse ici à la contribution de chaque variable individuellement. Pour chaque variable X_j, $1 \leq j \leq k$, on teste les hypothèses :

$$\begin{aligned} H_0 &: \beta_j = 0 \\ \text{contre } H_1 &: \beta_j \neq 0. \end{aligned}$$

Le non rejet de l'hypothèse nulle H_0 signifie que la variable X_j ne contribue pas au modèle. Deux méthodes sont disponibles pour effectuer le test. La première consiste à utiliser le rapport des vraisemblances comme pour le test global. Notons M le modèle complet et M_{-j} le modèle sans la variable X_j. La statistique du test s'écrit à l'aide des déviances des deux modèles :

$$LR = D_{M_{-j}} - D_M.$$

Lorsque l'hypothèse nulle H_0 est vraie, LR suit une loi de Khi-deux à un degré de liberté. Ainsi au seuil critique α, on rejette H_0 si $LR > \chi_{\alpha;1}$ ou si la p-valeur$=P(\chi_1^2 > LR) < \alpha$.

Dans la deuxième méthode, on utilise la statistique de Wald. La statistique de Wald repose sur la propriété de normalité asymptotique des estimateurs de maximum de vraisemblance. En effet pour une taille d'échantillon n grande, $\frac{\hat{\beta_j}}{s(\hat{\beta_j})}, j = 0, 1, \ldots, k$ suit approximativement une loi normale centrée réduite. L'erreur-type de l'estimateur $\hat{\beta}_j$ est $s(\hat{\beta}_j) = \sqrt{v_{jj}}$ où v_{jj} est tel que $V = (X^T C X)^{-1}$. La statistique du test de Wald est (Hosmer et al., 2013) :

$$W = \left(\frac{\hat{\beta}_j}{s(\hat{\beta}_j)} \right)^2. \tag{3.15}$$

Lorsque l'hypothèse H_0 est vraie, W suit une loi de Khi-deux à un degré de liberté. Le rejet de H_0 s'effectue alors si $W > \chi_{\alpha;1}$ au seuil critique α ou p-valeur=$P(\chi_1^2 > W) < \alpha$.

Le test global permet de s'assurer de la significativité de notre modèle. Les tests individuels quant à eux constituent un premier moyen de sélection des variables. Tous ces tests seront mis en œuvre dans l'analyse de nos données. Il existe différentes procédures de sélection de variables ; Hosmer et al. (2013) proposent une démarche en sept étapes. Dans notre projet, le nombre de variables est faible, ne nécessitant donc pas un important travail de sélection.

Nous avons présenté les deux cas extrêmes : soit on teste toutes les variables ensemble, soit une variable à la fois. Il existe aussi un test de rapport de vraisemblances ainsi qu'un test de Wald pour tester la contribution d'un sous-ensemble de variables.

Les pseudo-coefficients de détermination

Dans un modèle linéaire, le coefficient de détermination R^2 mesure le degré d'explication du modèle. Il est compris entre 0 et 1 et quantifie le pourcentage de variation totale expliquée par le modèle. En régression logistique il n'y a pas un consensus sur le coefficient de détermination. En effet plusieurs mesures existent dans la littérature. Les deux plus populaires sont ceux de McFadden R^2_{McF} et de Cox et Snell R^2_{CS}. Ces deux indices sont définis par (Allison, 2014) :

$$R^2_{McF} = 1 - \frac{\ell(M)}{\ell(M_0)}\ , \quad R^2_{CS} = 1 - \left(\frac{\mathcal{L}(M)}{\mathcal{L}(M_0)}\right)^{\frac{2}{n}}.$$

Contrairement au modèle linéaire, le coefficient de détermination ici ne mesure pas la proportion de variance expliquée mais plutôt une amélioration du modèle par rapport au modèle nul. Une valeur élevée de ces coefficients signifie que le modèle est globalement significatif.

3.1.4 Cas des données groupées

Dans le cas précédent, nous disposions de données binaires. Par exemple pour un étudiant donné, on s'intéresse à son résultat, succès ou échec, à chaque examen passé. Ici nous nous intéressons plutôt au nombre d'examens auxquels l'étudiant a réussi sur un nombre n d'examens passés ; on a donc le nombre de succès et le nombre de tentatives. On parle alors de données groupées.

Il s'agit du type de données que nous avons pour notre projet. En effet pour un mot clé nous avons le nombre de clics et le nombre d'impressions. Le nombre de clics représente le nombre de succès et le nombre d'impressions le nombre d'essais. Ainsi le taux de clics que nous cherchons à prédire correspond à la probabilité d'obtenir un succès.

Plus formellement, considérons n observations indépendantes de la forme $\{(\mathbf{x}_i, y_i), i = 1, 2, \ldots, n\}$. La variable $Y_i|\mathbf{x}_i$ est distribuée selon une loi binomiale de paramètres m_i et $p(\mathbf{x}_i)$, c'est-à-dire $Y_i|\mathbf{x}_i \sim \mathcal{B}in(m_i, p(\mathbf{x}_i))$ où m_i est le nombre de tentatives pour l'observation i. On remarquera que le cas binaire n'est que le cas particulier pour $m_i = 1$ pour tout $i = 1 \ldots, n$. Le cas des données groupées reste très similaire au cas binaire. En effet on s'intéresse toujours à la prédiction des probabilités. Le modèle *logit* est exactement le même que le modèle 3.1 ; l'analyse statistique (test global, test individuel, etc) est similaire. La différence réside dans l'expression de la fonction de vraisemblance à maximiser. En effet pour une loi binomiale la fonction de masse s'écrit :

$$p(y_i; \beta) = \binom{m_i}{y_i} p_i^{y_i} (1 - p_i)^{m_i - y_i}, \quad y_i \in \{0, 1, \ldots, m_i\},\ i = 1, \ldots, n. \tag{3.16}$$

La fonction de vraisemblance et la log vraisemblance sont alors :

$$\mathcal{L}(\beta; y_1, y_2, \ldots, y_n) = \prod_{i=1}^{n} \binom{m_i}{y_i} p_i^{y_i} (1 - p_i)^{m_i - y_i}, \tag{3.17}$$

$$\ell(\beta; y_1, y_2, \ldots, y_n) = \sum_{i=1}^{n} \log \binom{m_i}{y_i} + \sum_{i=1}^{n} \left[y_i \log(p_i) + (m_i - y_i) \log(1 - p_i) \right]. \tag{3.18}$$

La première somme dans l'expression de la log vraisemblance est une constante ; elle n'intervient donc pas dans la maximisation. Les équations (3.5) et (3.18) sont alors très similaires. L'équation (3.18) apparaît comme une version pondérée de l'équation (3.5). En somme les méthodes utilisées précédemment pour estimer les paramètres sont encore valides ici.

3.1.5 La régression logistique multinomiale

Ici la variable dépendante Y admet plus de deux catégories. À titre d'exemple, plutôt que de déterminer simplement si un étudiant réussit ou pas son examen, on peut le classer selon la lettre obtenue soit entre A, B, C, D et F. Dans cet exemple, on dispose alors de cinq catégories.

Considérons n observations indépendantes $\{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}$. On suppose qu'on dispose de C classes, c'est-à-dire que $Y_i \in \{1, \ldots, C\}$ et que chaque $\mathbf{x}_i$ est une réalisation du vecteur aléatoire $X = (X_1, X_2, \ldots, X_k)^T$ où chaque X_j, $i = 1, \ldots, k$ est une variable explicative.
La variable $Y_i|\mathbf{x}_i$ suit une loi multinomiale de paramètre $p_1(\mathbf{x}_i), \ldots, p_C(\mathbf{x}_i)$ où $p_c(\mathbf{x}_i) = P(Y = c|X = \mathbf{x}_i)$ est la probabilité d'appartenir à la classe c étant donné les valeurs des variables explicatives. On choisit une classe de référence (la classe C par exemple). Le modèle logistique

multiclasse se définit alors comme suit :

$$\log\left(\frac{p_c(\mathbf{x}_i)}{p_C(\mathbf{x}_i)}\right) = \beta_0^{(c)} + \beta_1^{(c)}x_{i1} + \beta_2^{(c)}x_{i2} + \ldots + \beta_k^{(c)}x_{ik} = \mathbf{x}_i^T\beta^{(c)}, \tag{3.19}$$

où $c \in \{1, \ldots, C-1\}$; $i = 1, \ldots, n$, $\beta_0^{(c)}, \beta_1^{(c)}, \ldots, \beta_k^{(c)}$ sont des paramètres réels et $\beta^{(c)} = (\beta_0^{(c)}, \beta_1^{(c)}, \ldots, \beta_k^{(c)})^T$.

Ainsi on retrouve les probabilités en utilisant la fonction sigmoïde :

$$p_c(\mathbf{x}_i) = \frac{\exp(\mathbf{x}_i^T\beta^{(c)})}{1 + \sum_{k=1}^{C-1}\exp(\mathbf{x}_i^T\beta^{(k)})}, \quad c = 1, \ldots, C-1 \tag{3.20}$$

$$p_C(\mathbf{x}_i) = \frac{1}{1 + \sum_{k=1}^{C-1}\exp(\mathbf{x}_i^T\beta^{(k)})}. \tag{3.21}$$

Le modèle de classification basé sur la régression logistique consiste alors à classer l'observation x_0 dans la classe l telle que

$$l = \underset{c\in\{1,\ldots,C\}}{\arg\max}\, \hat{p}_c(\mathbf{x}_0),$$

où $\hat{p}_c(\mathbf{x}_0)$, $c = 1, \ldots, C-1$ sont données par l'équation (3.20) avec des estimations $\hat{\beta}^{(c)}$ de $\beta^{(c)}$ et $\hat{p}_C(\mathbf{x}_0)$ est donnée par l'équation (3.21). Les estimations des paramètres $\beta^{(c)} = (\beta_0^{(c)}, \beta_1^{(c)} \ldots, \beta_k^{(c)})^T$ s'obtiennent par la méthode du maximum de vraisemblance. Notons $\beta = (\beta^{(1)}, \ldots, \beta^{(C-1)})^T$. La fonction de masse de la variable $Y_i|\mathbf{x}_i$ est

$$p(y_i;\beta) = \prod_{c=1}^{C} p_c(\mathbf{x}_i)^{\mathbb{1}_{\{y_i=c\}}} \quad \text{où} \quad \mathbb{1}_{\{y_i=c\}} = \begin{cases} 1 & \text{si } y_i = c \\ 0 & \text{sinon,} \end{cases} \tag{3.22}$$

et la vraisemblance est de la forme 3.4, précisément

$$L(\beta; y_1, y_2, \ldots, y_n) = \prod_{i=1}^{n} p(y_i;\beta).$$

Des méthodes d'optimisation disponibles dans la plupart des logiciels statistiques permettent d'obtenir l'estimateur du maximum de vraisemblance $\hat{\beta}$.

À l'image du cas de deux classes, il existe des tests statistiques permettant de vérifier le modèle globalement et de tester les coefficients individuellement. Nous disposons également de pseudo-coefficients de détermination. Nous ne présentons pas ces méthodes et métriques dans ce mémoire, car ils sont très similaires au cas binaire.

La régression logistique maintenant bien définie, nous présentons notre modèle de prédiction des taux de clics.

3.2 Le modèle de prédiction des taux de clics

Le taux de clics se définit comme le rapport du nombre de clics sur le nombre d'impressions. À chaque impression d'une annonce, l'utilisateur clique ou non sur celle-ci. Le taux de clics apparaît alors comme la probabilité d'un clic. On est donc en présence d'une épreuve de Bernoulli où le clic est le succès. Notons que lorsqu'une annonce s'affiche, cette impression est liée au mot clé (le plus similaire à la requête de l'utilisateur) que le moteur de recherche a choisi parmi ceux fournis par l'annonceur. Par exemple, considérons un magasin spécialisé en chaussures d'hiver et qui compte « chaussures extérieures » et « chaussures hiver neige » parmi ses mots clés. Son annonce apparaîtra pour des requêtes proches de ces deux mots clés. L'utilisateur, dont la requête est « chaussures extérieures », est peut être à la recherche de chaussures de randonnée, de bottes de jardins par exemple ; tandis que celui dont la requête comporte le second mot clé recherche effectivement des chaussures d'hiver. En raison notamment de l'indice de qualité calculé par le moteur de recherche, il est tout à fait raisonnable de penser que l'annonce n'apparaîtra pas à la même position suivant le mot clé ; ce qui conduit à des taux de clics différents pour les mots deux clés. À chaque impression de l'annonce, on admet donc que l'utilisateur clique dessus avec une probabilité p dépendant du mot clé. Si on suppose que chaque impression est propre à un utilisateur, on a raisonnablement une indépendance des impressions. On se retrouve alors avec une répétition d'épreuves de Bernoulli indépendantes et identiques. On obtient alors une distribution binomiale sur le nombre de clics avec comme paramètres n le nombre d'impressions et le taux de clics pour la probabilité p d'un succès. Les paramètres de cette distribution sont propres à chaque mot clé. Ce mécanisme décrit correspond exactement à celui du modèle logistique présenté à la section 3.1.1. De plus, le paramètre n de la loi binomiale, c'est-à-dire le nombre d'impressions, étant supérieur à 1, il s'agit donc d'un cas de données groupées.

Un modèle à retenir pour la prédiction des taux de clics est alors le modèle logistique dans le cas de données groupées. Néanmoins comme nous le montrons dans la suite, l'application directe de ce modèle ne donne pas de bons résultats dans le cas de nos données.

3.2.1 Motivation de l'approche avec double régression logistique

Les données dont nous disposons contiennent beaucoup d'observations à taux de clics nul ou unitaire. C'est ce qui motive notre approche avec une double régression logistique : la

première régression permet d'écarter ces observations et la seconde prédit le taux de clics pour les autres observations.

Dans les mots clés, on a généralement un fort pourcentage des observations avec un taux de clics nul (plus de 70% dans nos données) et quelques uns avec un taux égal à 1. Or le modèle logistique ne permet pas de prédire exactement les valeurs 0 et 1. En effet 0 et 1 sont les valeurs asymptotiques de la fonction sigmoïde et ne sont jamais atteintes (voir Figure 3.1). Ainsi les observations à taux de clics nul ou unitaire ne seront jamais correctement prédites. Toutefois, on pourrait se dire qu'une prévision exacte de la valeur zéro n'est pas importante tant que la valeur prédite reste très petite et proche de zéro. Effectivement nous pourrions nous satisfaire de cette prévision et fixer par exemple un seuil en dessous duquel une prévision sera considérée nulle et un autre seuil au dessus duquel la prévision sera ramenée à l'unité.

Cette méthode a donc été appliquée dans un premier temps : le modèle logistique est ajusté sur les données avec les variables explicatives à notre disposition à savoir : les mots clés (comme variable catégorique), la position moyenne et le coût. Le coût ici désigne la somme des coûts de tous les clics pour un mot clé et non un coût moyen. L'interaction entre les variables coût et position moyenne a été rajoutée dans le modèle, car elle était significative selon le test de Wald (équation 3.15). Une application succincte du modèle sur un de nos jeux de données montre effectivement que la valeur zéro n'est pas prédite mais plutôt des valeurs proches de zéro. Aussi, toutes les prédictions restent faibles puisque la valeur maximale prédite est 0,1381 sachant que les données contiennent des réalisations supérieures 0,2. L'idée ici consiste simplement à illustrer brièvement la problématique liée au fort pourcentage de valeurs nulles. Le chapitre 5 présentera en détails la procédure d'obtention de ces résultats.

La figure 3.2 permet de mieux constater ce phénomène ; les valeurs prédites sont tracées en fonction des valeurs réelles. En rouge (points ronds), nous avons la première bissectrice : idéalement les prédictions devraient suivre cette droite. Elle permet de visualiser la qualité des prédictions. Comme on peut le voir, on a bien beaucoup de points autour de l'origine, traduisant les petites valeurs prédites pour les taux de clics nuls. Néanmoins, on remarque aussi que pour les valeurs de taux de clics supérieures à 0,1, les points ne suivent plus la première bissectrice ; tous les points noirs (points carrés) restent très proches de 0 lorsque le taux de clics réel augmente. Ainsi nous sous-estimons toujours ces taux de clics.

La figure 3.3 représente toujours les taux de clics prédits en fonction des réels mais cette fois en retirant les observations à taux de clics nuls avant d'ajuster le modèle. Ici on voit que les prédictions décollent de zéro : les points noirs s'alignent à peu près le long de la bissectrice. On note surtout que cette fois, les valeurs prédites ne sont pas toujours inférieures aux valeurs réelles. Aussi les écarts par rapport à ces dernières augmentent avec le taux de clics

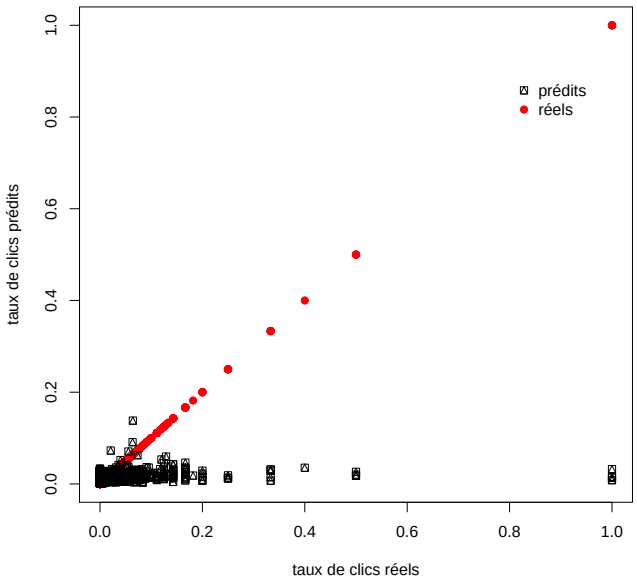

Figure 3.2 Nuage de points des taux de clics prédits en fonction des taux de clics réels.

car nous n'avons pas beaucoup d'observations pour les taux élevés, ce qui contribue ainsi à l'augmentation de la variance de nos prédictions. Enfin, on remarque un autre problème avec les taux de clics égaux à 1. En effet nous prédirons toujours des valeurs en dessous de 1.

Avec ces deux figures, on remarque que l'excès de valeurs nulles ne permet pas une bonne prédiction des taux de clics élevés qui en revanche sont relativement bien prédits en leur absence. L'idée est alors de réussir à écarter judicieusement les valeurs nulles ainsi que celles qui ont un taux égal à 1 pour ensuite ne considérer que les autres.

3.2.2 Le modèle

Pour une observation donnée, le but est d'abord de déterminer si elle a un taux de clics nul ou non. Si on considère que le taux est nul alors il sera prédit exactement à 0 et sinon le modèle logistique est utilisé pour prédire ce taux non nul. Nous avons donc deux étapes : une classification puis une régression.

Pour faire une classification, nous avons besoin d'étiquettes sur nos observations c'est-à-dire des classes. Or nous n'avons que le taux de clics qui est une variable continue en sortie. Une première classe qui apparaît naturellement est la classe "0" pour les observations à taux de clics nul. Ensuite nous avons la classe "1" pour les taux égaux à 1 et une troisième classe "2"

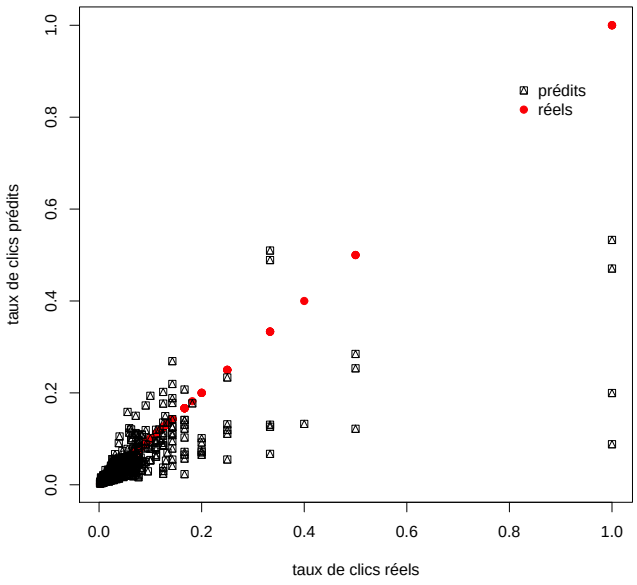

Figure 3.3 Nuage de points des taux de clics prédits en fonction des taux de clics réels pour l'ajustement sans les observations à taux nul.

pour les observations à taux de clics différents de 0 et 1. Cette dernière classe peut être éclatée en plusieurs classes. Nous en reparlons plus en détails dans le chapitre 5 sur les résultats. Pour les observations des classes "0" et "1", la prédiction du taux de clics est immédiate. Quant aux observations de la classe "2", nous ajustons le modèle logistique dans le cas des données groupées. Pour l'étape de classification, nous utilisons également la régression logistique mais cette fois dans le cas d'un classificateur multiclasse. Le modèle se résume alors comme suit :

1. Étiquetage des données ;
2. Classification pour écarter les 0 et les 1 en utilisant la régression logistique ;
3. Ajustement du modèle logistique pour les observations restantes : avec les observations à taux de clics compris entre 0 et 1 strictement, on ajuste un modèle logistique dans le cas des données groupées afin d'estimer le taux de clics.

3.2.3 Variables inconnues dans le modèle

Quelques problèmes demeurent pour pouvoir appliquer le modèle de prédiction des taux de clics. En effet des variables explicatives sont utilisées dans l'ajustement des modèles logistiques. Il s'agit de la position moyenne, du nombre d'impressions et du coût des clics. Dans

la méthode présentée plus haut, les données sur ces variables sont supposées toutes connues. Par exemple supposons un modèle qui estime le prix d'une maison en fonction de la superficie et du nombre de pièces. Pour connaître le prix d'une nouvelle maison, on fournit au modèle la superficie (40 m^2 par exemple) et le nombre de pièces (4 pièces par exemple). Dans notre cas, si on considère un mot clé à un jour futur, on ne connaît pas sa position moyenne, son nombre d'impression et son coût. Ainsi pour estimer le taux de clics pour ce mot clé, des estimations de la position moyenne, du nombre d'impressions et du coût sont nécessaires. Il nous faut donc d'abord déterminer des modèles pour estimer ces variables.

Supposons qu'on dispose des réalisations pour un ensemble de mots clés sur une certaine période (30 jours par exemple) jusqu'au jour t, c'est-à-dire un historique des positions moyennes réelles occupées, du nombre d'impressions réalisées et des coûts payés pour les clics pour chaque mot clé. Une modélisation « en chaîne » des variables est effectuée selon le schéma ci-contre :

1. $Position\,moyenne$: variable aléatoire de loi normale tronquée (détails à la section 4.1) ;
2. $Impression = f(Mot\,clé, Position)$;
3. $Coût = g(Mot\,clé, Position, Impression)$;
4. $CTR = h(Mot\,clé, Position, Impression, Coût)$.

Rappelons que si on considère une variable de réponse Y et une variable explicative X, ajuster un modèle à la variable Y consiste à trouver une fonction f telle que $Y \approx f(X)$. On remarquera que pour la position moyenne, il n'y a pas de fonction f. En effet, il s'agit de la première variable de la chaîne et elle n'utilise pas réellement de variables explicatives ; un modèle est ajusté pour chaque mot clé. Ainsi, la position moyenne est modélisée comme une variable aléatoire réelle ; pour un mot clé, l'historique de ses positions moyennes est utilisé pour déterminer les paramètres de la loi suivie par la position moyenne, en l'occurrence une loi normale tronquée à 1. Quant aux autres variables (nombre d'impressions et coût), contrairement à la position moyenne, un seul modèle est ajusté pour tous les mots clés. Pour le nombre d'impressions, le modèle utilise la position moyenne et les mots clés comme variables explicatives. Le modèle sur la variable coût quant à lui fait intervenir le nombre d'impressions en plus des deux variables précédentes. Enfin toutes ces variables sont utilisées dans le modèle sur le taux de clics. Le graphe de la figure 3.4 résume l'ordre de modélisation des variables. Une flèche entrante dans une boîte indique que la variable à la queue de la flèche est utilisée dans la modélisation de celle de la boîte. Par exemple, le mot clé et la position sont utilisés dans le modèle des impressions. Le chapitre 4 décrit en détails le modèle ajusté à chacune de ces variables.

Avec les données disponibles jusqu'au jour t, les fonctions f, g et h sont estimées de même

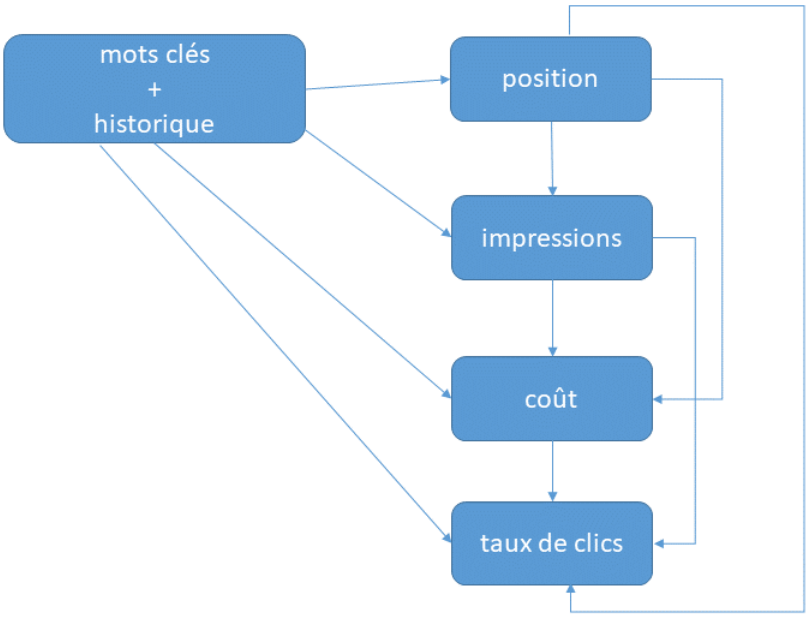

Figure 3.4 Graphe représentant la relation entre les différentes variables.

que les paramètres de la loi suivie par la position moyenne. Au jour $t+1$, seul le mot clé (son identifiant plus précisément, *key* par exemple) est connu. Ainsi, nous estimons d'abord la position moyenne $position(t+1)$ par une simulation de la loi normale tronquée. Ensuite le nombre d'impressions est prédit par

$$Impression(t+1) = f(key,\, Position(t+1)).$$

Puis ces deux estimations permettent d'estimer le coût :

$$Coût(t+1) = g(key,\, Position(t+1),\, Impression(t+1)).$$

Finalement la valeur d'intérêt, le taux de clics est estimée par

$$CTR(t+1) = g(key,\, Position(t+1),\, Impression(t+1),\, Coût(t+1)).$$

C'est ici qu'apparaît la notion de modélisation « en chaîne », puisque des estimations de

modèles sont réutilisées par d'autres modèles.

Le modèle logistique ainsi que les modèles que nous utilisons pour les variables intermédiaires ont la caractéristique commune de faire une hypothèse d'indépendance sur les observations. Or on peut raisonnablement, de prime abord, mettre en doute la validité de cette hypothèse dans notre cas puisque nous disposons de réponses sur les mêmes mots clés dans le temps. Dans la section suivante, nous montrons que cette hypothèse est heureusement vérifiée dans nos données.

3.3 Validation des hypothèses de travail

Beaucoup de modèles de régression ou de classification supposent l'indépendance des réalisations de la variable dépendante Y. En effet si on considère deux réalisations y_i et y_j de la variable Y, on suppose que celles-ci sont indépendantes. Dans nos analyses, nous utilisons des modèles tels que la régression logistique, la régression Poisson (pour les impressions) qui font également cette hypothèse.

L'idée que cette hypothèse ne serait pas vérifiée par nos données provient du mécanisme d'obtention des réponses. En effet, nous disposons d'un ensemble de mots clés qui sont envoyés aux moteurs de recherche ; ces derniers renvoient les résultats quotidiens de ces mots clés. Ainsi nous avons des mesures répétées sur les mots clés. Nous pouvons donc soupçonner une dépendance temporelle dans les réponses (nombre d'impressions, position, clics) puisqu'elles sont récoltées quotidiennement. Par exemple on peut se demander si le nombre d'impressions au jour $t+1$ dépend du nombre d'impressions du jour t. Les sujets ici sont les mots clés.

Il existe des méthodes complexes pour traiter ce type de données. Nous avons par exemple envisagé utiliser les équations d'estimation généralisées à la place de notre modèle logistique. Cependant une application de cette méthode sur nos données n'a pas semblé donner de meilleurs résultats comparativement à notre approche logistique ; au contraire, nous obtenions des résultats moins bons. Les métriques de comparaison sont détaillées dans la section résultats mais il s'agit essentiellement d'une erreur quadratique moyenne de prédiction. C'est ainsi que nous avons remis en doute la dépendance temporelle des données. Nous avons donc cherché à montrer qu'effectivement nous pouvons considérer nos observations indépendantes. Les techniques utilisées pour justifier cette indépendance sont issues de l'analyse des séries chronologiques. Nous montrons qu'au temps t chacune des variables : position moyenne, nombre d'impressions, coût et taux de clics est la somme d'un bruit blanc et d'une constante.

3.3.1 Quelques éléments d'analyse des séries chronologiques

Afin de montrer que nos observations peuvent être considérées comme indépendantes, nous utilisons quelques éléments de l'analyse des séries chronologiques.

Considérons une série d'observations dans le temps $y_1, y_2, \ldots, y_t$. On considère que y_t est indépendant du temps s'il peut s'écrire sous la forme (Brockwell et Davis, 2013)

$$y_t = \mu + w_t, \tag{3.23}$$

où $\{w_t, t \in \mathbb{Z}\}$ est un bruit blanc, c'est-à-dire une suite de variables non corrélées, de moyenne nulle et de même variance. Ci-dessous nous donnons la définition de quelques éléments utiles dont celle d'un bruit blanc.

Processus stochastique

Un processus stochastique est une famille de variables aléatoires de la forme $\{X_t, t \in T\}$. L'indice t s'interprète souvent comme le temps et l'ensemble T constitue l'espace temps du processus ; T peut être discret (exemple $T = \mathbb{N}$) ou continu (exemple $T = [0, \infty)$).

Série chronologique

Une série chronologique est une réalisation finie d'un processus stochastique.

Un outil permettant d'évaluer le degré de dépendance dans une série chronologique est la fonction d'autocovariance et surtout, la fonction d'autocorrélation. Ces fonctions sont définies comme suit.

Fonction d'autocovariance, fonction d'autocorrélation

Soient $\{X_t, t \in T\}$ une série chronologique et $s, t \in T$.
La fonction d'autocovariance de $\{X_t, t \in T\}$ se définit par

$$\gamma(s,t) = \mathrm{Cov}(X_t, X_s) = \mathbb{E}[(X_s - \mu_s)(X_t - \mu_t)] \tag{3.24}$$

avec $\mu_t = \mathbb{E}(X_t)$ la moyenne de la série chronologique.

La fonction d'autocorrélation se définit par

$$\rho(s,t) = \frac{\gamma(s,t)}{\sqrt{(\gamma(s,s)\gamma(t,t)}}. \tag{3.25}$$

Elle est souvent notée ACF dans la littérature. Une propriété intéressante des processus stochastiques est la stationnarité.

Stationnarité au sens strict et stationnarité au sens large

Un processus stochastique $\{X_t, t \in \mathbb{Z}\}$ est dit stationnaire au sens strict si la distribution de probabilité conjointe des variables $(X_{t_1}, X_{t_2}, \ldots, X_{t_l})$ est identique à celle des variables $(X_{t_1+h}, X_{t_2+h}, \ldots, X_{t_l+h})$ quels que soient les entiers l, t_1, ..., t_l et h.
Cette définition de la stationnarité est généralement trop forte à utiliser dans les applications. Une définition plus souple reposant sur les deux premiers moments est proposée.

Un processus stochastique est dit stationnaire au sens large si

- la moyenne du processus est constante c'est-à-dire qu'elle ne depend pas du temps : $\mu_t = \mu$;
- la variance du processus est finie : $\mathrm{Var}(X_t) < \infty$;
- Pour deux instants s et t, la fonction d'autocorrélation ne dépend que de la distance entre les deux instants : $\gamma(s,t) = \gamma(|s-t|)$.

On notera qu'un processus stationnaire au sens strict est stationnaire au sens large. La réciproque n'est pas toujours vraie sauf dans le cas d'un processus gaussien. Pour un processus stationnaire, les fonctions d'autocovariance et d'autocorrélation se réécrivent :

$$\gamma(h) = \mathrm{Cov}(X_{t+h}, X_t) \qquad \text{et} \qquad \rho(h) = \frac{\gamma(h)}{\gamma(0)}, \quad h = 0, \pm 1, \ldots. \tag{3.26}$$

On note que les deux fonctions sont paires et ne sont donc calculées en pratique que pour des valeurs de h positives.
Un exemple de série chronologique stationnaire au sens strict est le bruit blanc.

Bruit blanc

Un processus stochastique $\{w_t, t \in \mathbb{Z}\}$ est un bruit blanc si

- $\mathbb{E}(X_t) = 0, \ \forall t \in \mathbb{Z}$
- $\gamma(s,t) = \begin{cases} \sigma_w^2 & \text{si } s = t \\ 0 & \text{sinon} \end{cases}$ (3.27)

Il s'agit d'une suite de variables aléatoires non corrélées de moyenne nulle et de variance σ_w^2. Si ces variables sont de loi normale alors, elles sont indépendantes. On note qu'un bruit blanc est un processus stationnaire.

Différents modèles peuvent être ajustés à une série chronologique tels qu'une moyenne mobile, un modèle autorégressif ou encore le modèle mixte ARMA. Aucun de ces modèles ne nous intéressent réellement. Cependant il existe une méthode graphique assez classique permettant de sélectionner un modèle parmi les trois. Cette méthode repose sur le tracé de deux fonctions l'ACF et le PACF (que nous définissons plus bas). Ces deux fonctions permettent également d'identifier un bruit blanc.

Fonction d'autocorrélation partielle PACF

La fonction d'autocorrélation partielle notée ϕ_{hh} se définit comme la corrélation entre X_{t-h} et X_t compte tenu des variables intermédiaires $X_{t-h+1}, X_{t-h+2}, \ldots, X_{t-1}$. Si on note X_t^* et X_{t-h}^* les courbes de régression linéaire de X_t et X_{t-h} sur $X_{t-h+1}, X_{t-h+2}, \ldots, X_{t-1}$ alors

$$\phi_{hh} = \text{Corr}(X_t - X_t^*,\, Y_{t-h} - Y_{t-h}^*),\; h = 1, 2, \ldots. \tag{3.28}$$

Dans le cas d'un bruit blanc, on a $\phi_{hh} = 0$ pour tout $h = 1, 2, \ldots$. Pour son ACF, on a $\rho(0) = 1$ et $\rho(h) = 0$ pour $h = 1, 2, \ldots$. Ainsi dans la représentation graphique de l'ACF, nous avons simplement un pic en 0 qui atteint 1 et des 0 partout ailleurs et pour le PACF nous avons des 0 partout. La figure 3.5 représente l'ACF et le PACF sur une simulation de 100 observations d'un bruit blanc gaussien de variance $\sigma_w^2 = 1$. Sur l'ACF à gauche, seule l'autocorrélation en 0 est significative. Vu qu'il s'agit d'une simulation nous n'avons pas exactement des valeurs nulles pour les autres h. Toutefois aucune de ces autocorrélations ne dépassent les pointillés bleus qui représentent un intervalle de confiance. De même pour le PACF à droite, aucune des autocorrélations partielles ne dépassent les limites de confiance. Ainsi pour nos variables explicatives que sont la position, les impressions et les coûts, nous nous attendons à des graphiques d'ACF et de PACF similaires.

Tests de type *portemanteau*

Dans l'ajustement d'un modèle en séries chronologiques, une analyse des résidus est également faite à l'image de celle du modèle linéaire. Ici on cherche à s'assurer que les résidus sont bien un bruit blanc. En plus de l'analyse des graphiques de l'ACF et du PACF, d'autres tests sont également effectués notamment les tests de type *portemanteau* dont les plus populaires sont ceux de Box-Pierce et de Ljung-Box. Dans l'ACF par exemple, les autocorrélations sont analysées individuellement : on regarde si une valeur particulière d'autocorrélation est dans les limites d'acceptation (les pointillés bleus sur le graphique). Ici l'idée consiste à considérer

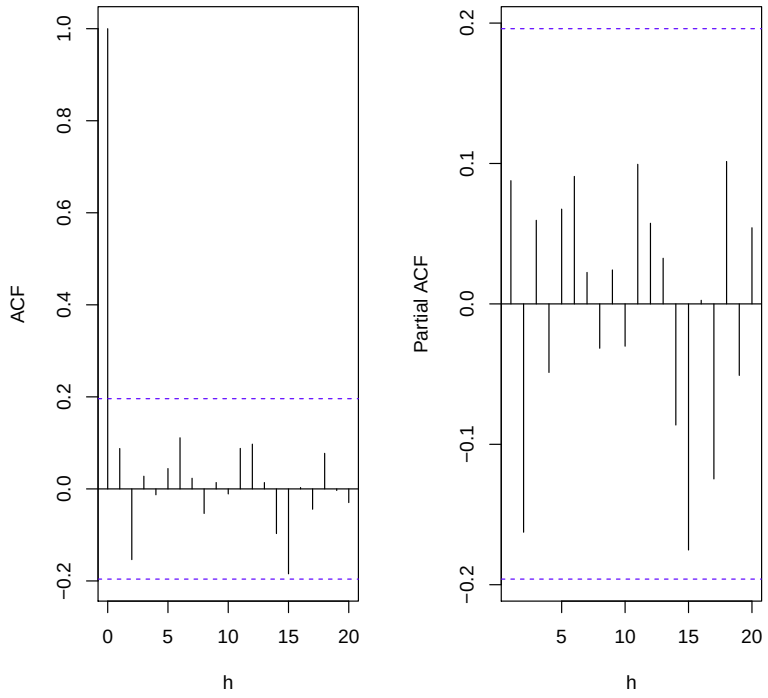

Figure 3.5 Graphique de l'ACF (à gauche) et du PACF (à droite) d'une simulation d'un bruit blanc gaussien.

les K premières valeurs des autocorrélations ensemble. On teste les hypothèses :

$$\begin{aligned} H_0 &: \text{les données sont non corrélées} \\ \text{contre } H_1 &: \text{les données sont corrélées.} \end{aligned}$$

Les statistiques des tests de Box-Pierce et Ljung-Box sont respectivement (Shumway et Stoffer, 2010) :

$$Q_{BP} = n \sum_{h=1}^{K} \hat{\rho}(h)^2, \qquad Q_{LB} = (n+2) \sum_{h=1}^{K} \frac{n}{n-h} \hat{\rho}(h)^2, \tag{3.29}$$

où n est la taille de l'échantillon et $\hat{\rho}(h)$ est l'estimateur de la fonction d'autocorrélation $\rho(h)$ à l'horizon h. La valeur de K est choisie arbitrairement ; souvent on prend $K = 20$.

Lorsque l'hypothèse H_0 est vraie, Q_{BP} et Q_{LB} suivent tous les deux une loi de Khi-deux à K degré de liberté. Ainsi au niveau critique α, on rejette H_0 si $Q_{BP} > \chi^2_{\alpha,K}$ pour le test de Box-Pierce. Pour le test de Ljung-Box, on rejette également l'hypothèse nulle si $Q_{LB} > \chi^2_{\alpha,K}$.

3.3.2 Application à nos variables

Les variables dont nous disposons sont : la position moyenne, le nombre d'impressions, le coût, le nombre de clics, le taux de clics. Ce dernier est le rapport du nombre de clics et du nombre d'impressions ; ainsi si les deux termes du rapport ne dépendent pas du temps alors le taux de clics est également indépendant du temps. Il n'est donc pas traité.

Nous considérons maintenant ces variables comme des séries chronologiques. L'unité de temps est la journée. Premièrement nous devons nous assurer que ces séries sont stationnaires. La représentation graphique d'une série chronologique est généralement suffisante pour vérifier si elle est stationnaire ou non. En effet la stationnarité se traduit par une moyenne constante qui n'évolue pas en fonction du temps et aussi une variance finie qui n'évolue pas en fonction du temps. Il existe des tests stationnarité tels que le test de Dickey–Fuller (Dickey et Fuller, 1979) ou encore le test de KPSS (Kwiatkowski et al., 1992) pour vérifier formellement la stationnarité. Cependant puisque l'hypothèse nulle sera rejetée par les tests *portmanteau* en cas de non stationnarité de la série, nous ne sommes pas attardés sur ces tests.

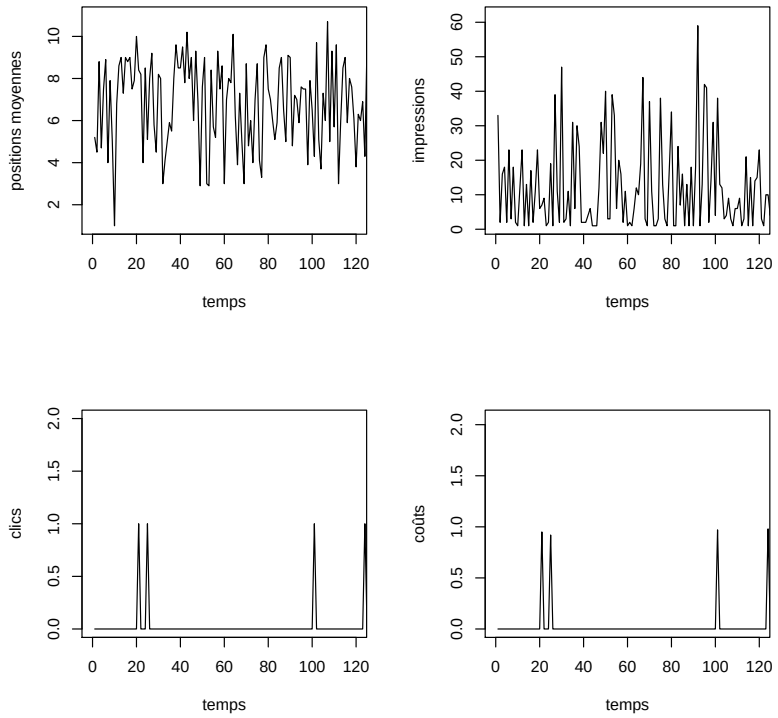

Figure 3.6 Évolution temporelle des positions, des impressions, des clics et des coûts pour un mot clé donné.

À titre d'illustration, les graphiques de la figure 3.6 représentent la position moyenne, le nombre d'impressions, le nombre de clics et le coût pour un mot clé donné en fonction du temps sur une période de 120 jours. Comme on peut le remarquer la moyenne ne semble pas augmenter ou diminuer en fonction du temps dans les quatre graphiques. Aussi il n'y a pas de présence de tendance particulière. Toutefois sans surprise, on constate beaucoup de valeurs nulles pour les clics et les coûts. On notera qu'on considère un mot clé en particulier, car nous supposons que la position (ou le nombre d'impressions ou le coût) d'un mot clé est indépendante de celle d'un autre mot clé ; c'est l'indépendance des répétitions sur un même mot clé qui nous intéresse. Nous avons donc examiné les graphiques de différents autres mots clés (50 mots clés) et les séries semblent toutes stationnaires au sens large. Cependant quelques rares séries ont semblé ne pas être stationnaires.

Nous traçons maintenant les graphiques de l'ACF et du PACF pour ces données. Les figures 3.7 et 3.8 représentent l'ACF et le PACF des séries chronologiques de la position moyenne et du nombre d'impressions représentées plus haut. Comme on peut le voir, pour l'ACF, nous obtenons une autocorrélation égale à un à l'horizon 0 et aucune des autres autocorrélations n'est significative. En effet pour un bruit blanc, d'après les équations (3.26) et (3.27), $\rho(0) = 1$ et $\rho(h) = 0$ pour $h = 1, 2, \ldots$. Sur le graphique du PACF, aucune des autocorrélations partielles n'est significative ; ce qui est normal puisque pour un bruit blanc, $\phi_{hh} = 0$ pour tout $h = 1, 2, \ldots$ (voir équation 3.28). Nous obtenons des graphiques similaires pour le nombre de clics et le coût. On peut noter une autocorrélation partielle à l'horizon $h = 17$ qui est significative pour le nombre d'impressions. Effectivement nous aurons de temps en temps des corrélations très légèrement significatives. L'idée c'est qu'elles le soient très légèrement comme c'est le cas ici. Enfin les tests de Box-Pierce et Ljung-Box ne rejettent pas l'hypothèse H_0 c'est-à-dire que les données ne sont pas corrélées.

Nous considérons 500 mots clés différents. Pour chacun de ces mots, les tests de Box-Pierce et Ljung Box sont effectués sur les positions moyennes, le nombre d'impressions et les coûts. Le paramètre K dans les équations 3.29 est fixé à 20 et le seuil critique α est choisi égal à 0,01. Pour la position moyenne, l'hypothèse H_0 est rejetée respectivement 183 et 189 fois avec le test de Box-Pierce et celui de Ljung-Box. Quant aux impressions, le nombre de rejet s'élève à 185 avec Box-Pierce et à 193 pour Ljung-Box. Enfin pour les coûts, avec le test de Box-Pierce, on rejette H_0 46 fois et 48 fois avec celui de Ljung-Box. Les résultats sont assez satisfaisants puisque nous constatons que l'hypothèse H_0 est majoritairement acceptée. L'hypothèse d'indépendance temporelle est donc plausible. Ainsi, pour la suite, les données seront considérées indépendantes et les méthodes supposant l'indépendance des observations seront appliquées.

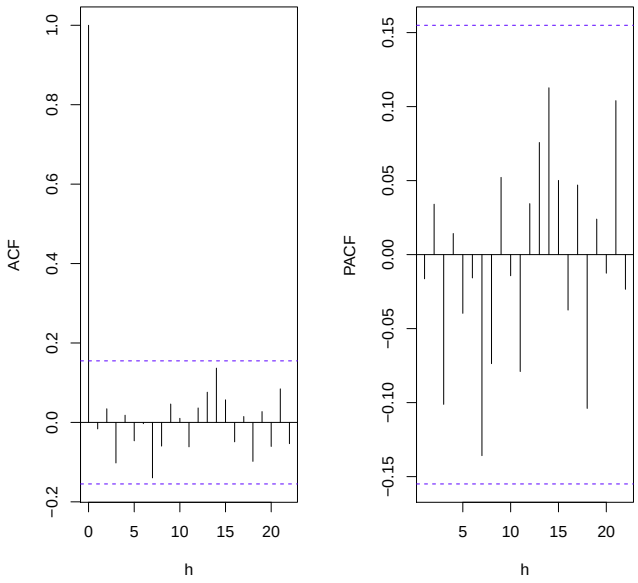

Figure 3.7 ACF et PACF de la série de la position moyenne représentée sur la figure 3.6.

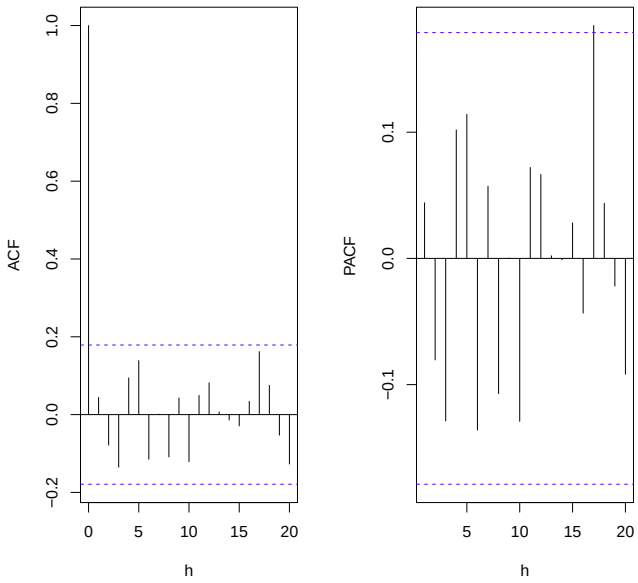

Figure 3.8 ACF et PACF de la série du nombre d'impressions représentée sur la figure 3.6.

La structure du modèle de prédiction des taux de clics est maintenant définie (section 3.2.2). Nous avons montré que nos variables pouvaient être raisonnablement considérées indépendantes. Nous présentons maintenant les données sur lesquelles le modèle est testé.

3.4 Présentation des données disponibles

Pour notre projet de mémoire, nous disposons de trois jeux de données que nous désignons par A, B et C. Ces données sont issues de véritables campagnes publicitaires effectuées par des annonceurs. Ainsi, elles sont confidentielles et le manque de précision ou de clarté sur celles-ci est fait sciemment. De plus, aucun travail d'extraction sur un serveur ou de requêtes SQL n'a été nécessaire puisqu'elles nous ont été fournies au format *.csv* et prêtes à être manipulées.

Les données A et B sont similaires ; elles représentent chacune les observations des mots clés d'une seule campagne sur une période de quatre mois. Le jeu de données A contient 33 mots clés et le jeu B 65. Ces deux jeux de données ont longtemps été les seules à notre disposition. Elles ont donc beaucoup déterminé le choix de nos méthodes. Le jeu de données C aura surtout servi à confirmer les hypothèses émises lors de l'analyse des deux premiers. Il contient environ 600000 entrées sur plus de 800 campagnes et environ 24000 mots clés distincts. Les mots clés sont observés entre une semaine et quatre mois. Nous avons retiré les mots clés utilisées moins de 3 mois afin d'avoir assez d'observations sur chaque mot clé.

Les données présentent deux types d'informations. D'une part, nous avons les caractéristiques du mot clé : son identifiant, son expression, la campagne à laquelle il appartient, le groupe de mots clés, la date de l'apparition de l'annonce associée, le type de correspondance, le statut du mot clé, le max CPC (uniquement pour les données A et B), l'url de l'annonce, le budget de la campagne. Peu de ces caractéristiques sont utilisables dans un modèle statistique. La plupart sont des facteurs emboîtés ou alors ne varient pas d'un mot clé à un autre. Par exemple le max CPC est une quantité intéressante mais elle est identique pour tous les mots clés. En effet dans les données dont nous disposons, sa valeur est la même pour tous les mots clés et ne varie pas dans le temps. Aussi le budget n'est disponible que dans les jeux de données A et B et à l'exception de sa valeur, on ne sait pas s'il est restrictif ou encore s'il est toujours atteint. Ainsi seul l'identifiant du mot clé est utilisé comme une variable catégorique. L'identifiant d'un mot clé est le code, unique, désignant ce mot clé. Quand nous parlons du mot clé comme variable, nous faisons référence en réalité à son identifiant.
D'autre part, nous avons les réalisations des mots clés à savoir le nombre d'impressions, le nombre de clics, le coût, le taux de clics, le nombre de conversions. Ce dernier est quasiment tout le temps nul. Comme nous le disions dans l'introduction, le taux de conversion est en

réalité la vraie valeur d'intérêt mais elle est presque tout le temps nulle rendant son analyse impossible.
Nous retirons les variables non utilisées de nos données et obtenons la structure type suivante :

Tableau 3.2 Structure type des données.

	MOTCLE_ID	POSITIONMOYENNE	IMPRESSIONS	CLICS	CTR	COÛTS
1	mot clé 1	1	1	0	0	0
2	mot clé 1	1,67	3	1	0,333	1,10
3	mot clé 2	2,60	92	0	0	0
4	mot clé 3	3,30	3	0	0	0
5	mot clé 3	6,60	5	0	0	0
6	mot clé 4	5,00	2	0	0	0
7	mot clé 5	6,50	13	1	0,07692	0,46
8	mot clé 5	6	1	0	0	0
9	mot clé 6	4,40	15	0	0	0
10	mot clé 7	7,40	7	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Notre méthode de prédiction de taux de clics a été développée avec les données A et B qui ont environ 3500 entrées. Cette méthode n'est pas applicable au jeu de données C à cause du grand nombre de mots clés qu'il contient. En effet l'ajustement de notre modèle échoue lorsque le nombre de mots clés est important. Nous en parlons en détails au chapitre 6 dans les limitations de notre solution (section 6.2). Ainsi nous créons de plus petits jeux de données à partir de celui-ci en tirant aléatoirement un certain nombre de mots clés (une trentaine en général). Nous considérons des mots clés et récupérons leurs observations dans le jeu C. Cette technique nous permet de créer un grand nombre de jeux de données différents. Ce qui nous permet de vérifier empiriquement certaines hypothèses.

Dans le chapitre 4, nous présentons en détails les modèles ajustés aux variables explicatives que sont la position moyenne, le nombre d'impressions et le coût, nécessaires à l'application du modèle de prédiction des taux de clics.

CHAPITRE 4 MODÉLISATION DES VARIABLES EXPLICATIVES

La position moyenne, le nombre d'impressions et le coût sont les variables explicatives de notre modèle double logistique. Dans un modèle, les données sur la variable explicative sont supposées connues pour toutes les observations. Par exemple si nous disposons d'un modèle qui calcule le prix d'une maison en fonction de sa surface, pour une nouvelle maison dont nous souhaitons connaître le prix nous connaissons et fournissons la valeur de la surface dans le modèle. Or ici pour un mot clé donné, pour une nouvelle période, nous ne connaissons pas sa position moyenne, son nombre d'impressions et son coût. Nous avons donc besoin de modèles afin d'estimer les valeurs de ces variables pour ensuite prédire le taux de clics.

4.1 La position moyenne

Pour modéliser la variable position moyenne, nous disposons de ses réalisations antérieures et des mots clés. Chaque mot clé (*Keyword_id*) présente des caractéristiques telles que le nombre de mots qu'il contient, la campagne à laquelle il appartient (*Campaign_id*), le groupe de mot clé auquel il est associé (*Adgroup_id*). Le nombre de mots n'étant pas disponible pour tous les jeux de données, il a donc été écarté. Les trois autres variables sont des variables catégoriques. Malheureusement elles sont emboîtées ou hiérarchiques. En effet un mot clé donné appartient à un unique groupe ; ce groupe à son tour appartient à une seule campagne. Nous ne pouvons donc pas utiliser les trois variables dans un modèle linéaire ou un modèle d'analyse de variance à cause de cette relation entre les variables : un problème de multicolinéarité se pose.

Il existe des méthodes permettant de tenir compte de l'emboîtement des facteurs. Toutefois elles s'intéressent principalement à l'effet des facteurs sur la réponse tandis que nous sommes plutôt intéressés par la prédiction. Ainsi nous choisissons d'écarter la campagne (*Campaign_id*) et le groupe (*Adgroup_id*) pour ne retenir que les mots clés (*Keyword_id*).

4.1.1 Pertinence des mots clés

La figure 4.1 représente les boîtes à moustache de la position moyenne pour différents mots clés. On remarque que les boîtes ne sont pas toutes identiques ; les quatre quartiles ainsi que la taille des moustaches varient d'un mot clé à un autre. Ce qui conforte l'idée que les valeurs de la position moyenne dépendent du mot clé.

Une analyse de variance est ensuite effectuée afin de faire un test sur l'égalité des moyennes

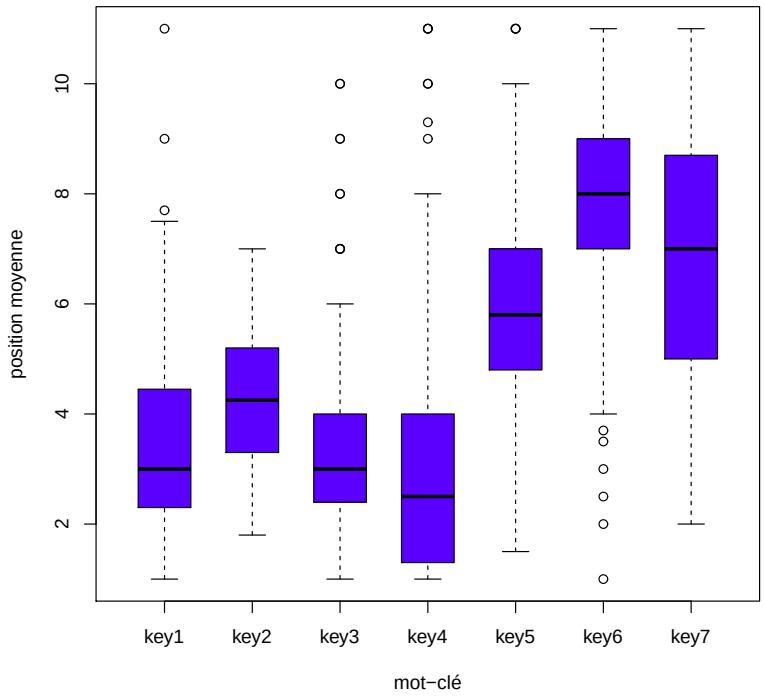

Figure 4.1 Diagramme de Tukey de la position pour différents mots clés.

de la position en fonction des mots clés. L'analyse de variance est une méthode d'analyse statistique permettant de comparer et de tester les moyennes de plus de deux populations. Il s'agit d'évaluer l'effet d'une variable appelée facteur sur une variable réponse quantitative. Dans notre cas, la variable de réponse Y est la position moyenne et le facteur est l'identifiant du mot clé qui possède $\boldsymbol{a}$ niveaux ou catégories. Les niveaux sont les différents mots clés. Par exemple pour une campagne contenant 30 mots clés distincts, on a 30 catégories, c'est-à-dire que $\boldsymbol{a}$=30. Puisqu'il n'y a qu'un facteur (une variable explicative), on parle alors de modèle d'analyse de variance à un facteur. Nous voulons tester si la position moyenne diffère selon les mots clés. On dispose des mesures de la position moyenne sur 120 jours pour les différents niveaux. Les hypothèses suivantes sont alors testées :

$$\begin{aligned} H_0 &: \mu_1 = \mu_2 = \ldots = \mu_a \\ \text{contre } H_1 &: \text{au moins deux des } \boldsymbol{a} \text{ moyennes sont différentes} \end{aligned}$$

Le rejet de l'hypothèse nulle H_0 signifie qu'au moins 2 mots clés ont des positions moyennes différentes. L'hypothèse H_0 a été rejetée pour l'ensemble de nos jeux de données. Aussi des

méthodes de comparaisons multiples sont utilisées pour identifier les mots clés différents. Ces méthodes permettent de comparer chaque couple de mots clés à un seuil critique α donné. Il existe plusieurs méthodes mais c'est la procédure de Tukey (HSD ou *Honest Significant Difference*) qui a été retenue. On remarque essentiellement que la plupart des mots clés sont distincts, c'est-à-dire qu'ils ont des positions moyennes différentes.
Ci-dessous nous présentons un exemple illustratif de l'analyse de variance.

Exemple : On dispose des positions moyennes de 4 mots clés pendant 7 jours. On a donc 4 niveaux pour lesquels 7 mesures de position moyenne sont disponibles pour chacun. Le tableau 4.1 illustre la structure des données de cet exemple.

Tableau 4.1 Tableau des données pour un modèle d'analyse de variance à un facteur.

Niveau	Mesures de position moyenne						
mot clé 1	9.5	4	6	3	3	5	3
mot clé 2	4	3,8	3,7	5	6	5,1	6,5
mot clé 3	1	1	1	3	2,5	2,3	2
mot clé 4	7	8	6,9	7,5	6,9	4,9	6,1

Le test d'égalité des moyennes est ensuite effectué. Ci-dessous la sortie du logiciel R pour les données du tableau 4.1.

```
my_anova <- aov(position~niveau, data=dt)
summary(my_anova)
Df Sum Sq Mean Sq F value   Pr(>F)
niveau       3  87,05  29,018   13,54 2.24e-05
Residuals   24  51,41   2,142
```

Nous obtenons une p-valeur égale à $2,24 \times 10^{-5}$. Au seuil critique $\alpha = 0,05$, $2,24 \times 10^{-5} < \alpha$, donc l'hypothèse H_0 est rejetée. Nous comparons ensuite les mots clés deux à deux par la procédure de Tukey. Nous obtenons la sortie R suivante :

```
TukeyHSD(my_anova)
diff        lwr        upr     p adj
mot2-mot1  0,08571429 -2,0724891  2,2439177 0,9995155
mot3-mot1 -2,95714286 -5,1153463 -0,7989394 0,0047481
mot4-mot1  1,97142857 -0,1867748  4,1296320 0,0820812
mot3-mot2 -3,04285714 -5,2010606 -0,8846537 0,0036350
```

```
mot4-mot2  1,88571429 -0,2724891  4,0439177 0,1020383
mot4-mot3  4,92857143  2,7703680  7,0867748 0,0000093
```

Les p-valeurs de chaque test de comparaisons sont données dans la colonne *p adj*. On remarque donc que les mots clés 1 et 2 sont très similaires (une p-valeur très élevée) tandis que les autres sont assez distincts.

Disposant uniquement d'une variable catégorique, nous ne pouvons pas ajuster des modèles complexes sur la position moyenne. Ci-dessous, deux méthodes simples sont comparées : l'analyse de variance et une approche probabiliste.

4.1.2 Approche par régression linéaire

Le modèle d'analyse de variance à un facteur est identique à un modèle de régression multiple. On introduit $\boldsymbol{a} - 1$ variables indicatrices $X_1, X_2, \ldots, X_{a-1}$ telles que

$$X_i = \begin{cases} 1 & \text{si mot clé } \boldsymbol{i} \\ 0 & \text{sinon.} \end{cases}$$

On obtient alors le modèle linéaire :

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_{a-1} X_{a-1} + \epsilon,$$

où les β_i, $i = 1, \ldots, \boldsymbol{a} - 1$ sont des paramètres réels et $\epsilon \sim \mathcal{N}(0, \sigma^2)$ une variable aléatoire.

Les paramètres β_i sont estimés avec la méthode des moindres carrés ordinaires. De plus nous obtenons en moyenne un coefficient de détermination supérieur à 60%. Ainsi le mot clé explique plutôt bien la variable position moyenne.

Avec cette approche, nous avons des prédictions constantes pour toute nouvelle observation d'un mot clé. En effet un mot clé $\boldsymbol{i}$ sera toujours prédit par $\bar{Y}_{i\cdot}$ soit la moyenne des positions de ce mot clé dans l'ensemble d'apprentissage. Une alternative à cette approche est l'approche probabiliste.

4.1.3 Approche probabiliste

Ici nous cherchons à déterminer la distribution de la variable position moyenne. Les positions que nous avons sont des moyennes journalières : le mot clé apparaît plusieurs fois au cours de la journée et nous récupérons la moyenne de ces positions. Si on suppose qu'un mot clé apparaît au moins 30 fois dans une journée, on pourrait alors appliquer le théorème central

limite et dire que chaque position moyenne suit approximativement une loi normale.

En outre la valeur d'une position est toujours supérieure ou égale à 1. Nous pouvons alors affiner le résultat précédent et considérer plutôt que la position moyenne suit une loi normale tronquée à 1. Par ailleurs nous avons montré que la position variait selon le mot clé. Ainsi les paramètres de la loi normale tronquée varient suivant le mot clé.

Supposons que nous avons $\boldsymbol{a}$ mots clés distincts. Pour chaque mot clé $\boldsymbol{i}$, on fait l'hypothèse que la variable aléatoire X_i désignant la position du mot clé $\boldsymbol{i}$ suit une loi normale tronquée à 1, de paramètres μ_i et σ_i^2 respectivement la moyenne et la variance de la loi normale sous-jacente.

Loi normale tronquée

Soient X une variable aléatoire de loi normale $\mathcal{N}(\mu, \sigma^2)$ et $]a, b[\subset \mathbb{R}$ un intervalle. La troncature de X à l'intervalle $]a, b[$ suit la loi conditionnelle $X|a \leq X \leq b$.
$X_{]a,b[} = X|a \leq X \leq b$ est alors dite de loi normale tronquée à l'intervalle $]a, b[$. La fonction de densité de $X_{]a,b[}$ est $g(x)$ définie par :

$$g(x) = \frac{f(x)}{P(a < X < b)},\ x \in [a, b], \quad \text{où} \quad f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{4.1}$$

et f est la fonction de densité de X. $P(a < X < b) = \Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})$ est la probabilité que X soit compris dans l'intervalle $]a, b[$ et Φ est la fonction de répartition de la loi normale centrée réduite. Dans le cas qui nous intéresse c'est-à-dire une loi normale tronquée à gauche en 1 ($X_{]1,\infty[}$), on a $a = 1$ et $b = \infty$. Ainsi la fonction de densité se réécrit :

$$g(x) = \frac{f(x)}{1 - \Phi(\frac{1-\mu}{\sigma})}. \tag{4.2}$$

La figure 4.2 représente la fonction g pour différentes valeurs de la moyenne et de la variance. Comme on peut le voir, la loi normale tronquée peut prendre plusieurs formes différentes. Par exemple pour $\mu = -2$ et $\sigma = 5$, nous obtenons une densité qui ressemble à celle d'une exponentielle. Ce qui permet de modéliser des positions telles que celles présentées sur les histogrammes en haut à droite et en bas à gauche sur la figure 4.3. De plus lorsque la moyenne est très supérieure à 1 nous obtenons des histogrammes en forme de cloche. Par contre pour une moyenne petite proche de 1, nous n'avons qu'une partie de la cloche.

On peut montrer que l'espérance et la variance d'une loi normale tronquée à gauche en 1

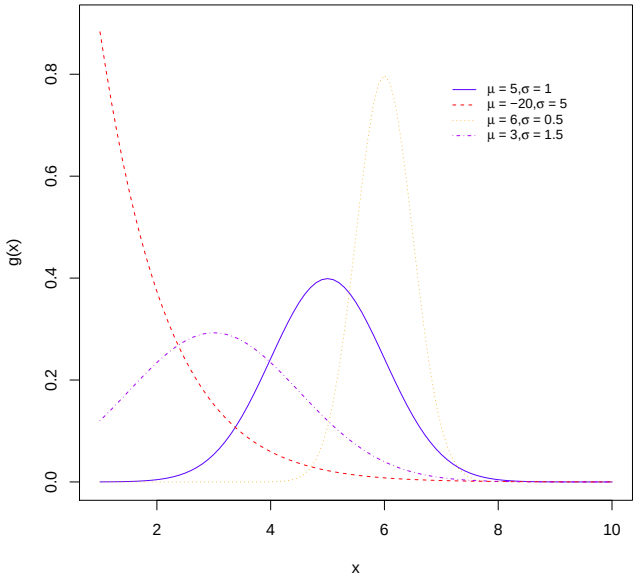

Figure 4.2 Graphe de la fonction de densité d'une loi normale tronquée à gauche en 1 pour différentes valeurs de μ et σ.

sont :

$$\mathbb{E}(X_{]1,\infty[}) = \mathbb{E}(X|X>1) = \mu + \frac{\sigma^2 f(1)}{1-\Phi(\frac{1-\mu}{\sigma})}, \tag{4.3}$$

$$\mathrm{Var}(X_{]1,\infty[}) = \mathrm{Var}(X|X>1) = \sigma^2\left[1 + \frac{f(1)}{1-\Phi(\frac{1-\mu}{\sigma})}\left(1-\mu-\frac{\sigma^2 f(1)}{1-\Phi(\frac{1-\mu}{\sigma})}\right)\right]. \tag{4.4}$$

Estimation des paramètres

La méthode de vraisemblance maximale est utilisée pour estimer les paramètres μ_i et σ_i^2 pour chaque mot clé $\boldsymbol{i}$. On suppose qu'on dispose de n_i observations $y_1, y_2, \ldots, y_{n_i}$ indépendantes de la position moyenne pour le mot clé $\boldsymbol{i}$. La fonction de vraisemblance s'écrit (Burkardt, 2014) :

$$\mathcal{L}(\mu_i, \sigma_i^2\,;\, y_1, y_2, \ldots, y_{n_i}) = \prod_{j=1}^{n_i} \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left(-\frac{(y_j-\mu_i)^2}{2\sigma_i^2}\right) \times \frac{1}{1-\Phi(\frac{1-\mu_i}{\sigma_i})}. \tag{4.5}$$

Le terme $\Phi(\frac{1-\mu_i}{\sigma_i})$ est une intégrale dont la valeur dépend des paramètres μ_i et σ_i^2. Surtout elle n'admet pas d'expression exacte. Ainsi les composantes du vecteur gradient ainsi que ceux de la matrice hessienne font intervenir des calculs d'intégrales. Les expressions exactes sont

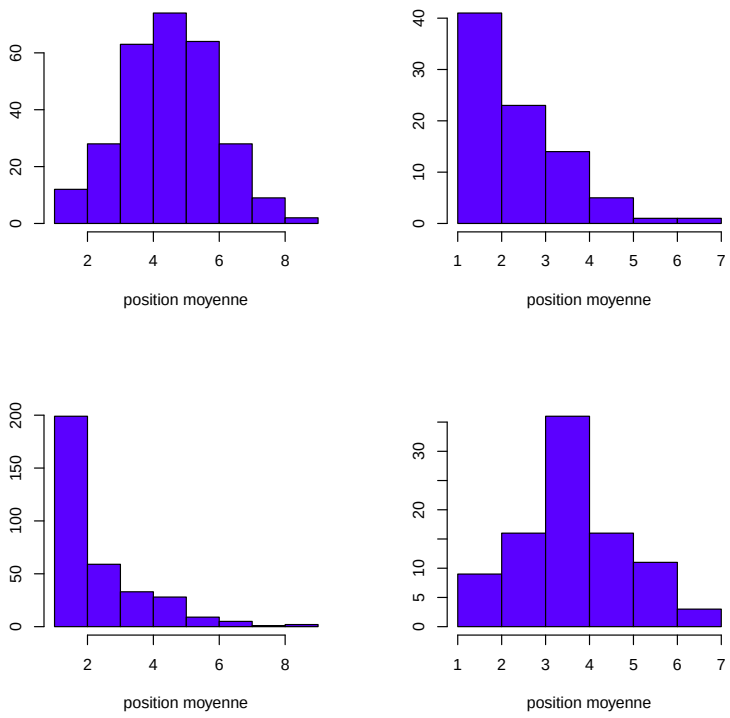

Figure 4.3 Histogramme de la position moyenne pour différents mots clés.

données dans (Hattaway, 2010). Néanmoins, pour simplifier, nous avons choisi de maximiser la vraisemblance en utilisant la méthode BFGS (Nocedal et Wright, 2006) et l'approximation des différences finies pour le gradient. Une estimation des paramètres par la méthode des moments est également possible. Néanmoins en raison des bonnes propriétés de l'estimateur de vraisemblance maximale, nous ne considérons pas cette méthode.

Nous avons utilisé notre connaissance sur la position ainsi que les histogrammes pour proposer la loi normale tronquée comme distribution. Maintenant que nous savons estimer les paramètres de la loi, nous pouvons utiliser des tests d'adéquation pour vérifier notre hypothèse de normalité tronquée. Les tests considérés sont ceux de Anderson-Darling et Cramer-von Mises.

Tests d'adéquation

Un test d'adéquation d'un échantillon $(x_1, \ldots, x_n)$ à une loi de probabilité donnée consiste à vérifier s'il est vraisemblable que x_1, x_2, …, x_n soient des réalisations de variables aléatoires

indépendantes X_1, X_2, ..., X_n de cette loi.

Notons F la fonction de répartition inconnue de l'échantillon et F_0 celle de la loi de probabilité considérée. On teste les hypothèses :

$$H_0 \, : \, F = F_0 \quad \text{contre} \quad H_1 \, : \, F \neq F_0.$$

Dans le cas de lois continues, les tests généralement utilisés sont ceux de Kolmogorov-Smirnov, Cramer-von Mises et le test d'Anderson-Darling. En raison du grand nombre d'ex-æquo, c'est-à-dire de valeurs égales, le test de Kolmogorov-Smirnov est écarté. Ainsi nous utilisons les deux autres tests pour vérifier notre hypothèse. L'idée principale de ces tests consistent à mesurer l'écart entre F et F_0 et de rejeter H_0 si cet écart dépasse un certain seuil (Gaudoin, 2013).

Nous considérons le jeu de données A contenant 33 mots clés et correspondant à une campagne. Nous estimons les paramètres de la loi normale tronquée $\{(\mu_i, \sigma_i^2), i \in \text{Keywords}\}$ (Keywords désigne l'ensemble des mots clés) pour chaque mot clé et nous effectuons les tests d'adéquation au seuil critique $\alpha = 0,05$. Cramer-von Mises rejette H_0 une seule fois et Anderson-Darling cinq fois. Ainsi notre choix semble plutôt judicieux puisque l'hypothèse H_0 est acceptée dans la plupart des cas.

On peut noter que lorsqu'un mot clé n'a essentiellement que des positions hautes (inférieures à 3 par exemple) la loi normale tronquée n'est plus un bon modèle. En considérant aléatoirement des mots clés de notre grand jeu de données, l'hypothèse nulle est aussi généralement acceptée par les tests. Pour les cas de rejet, on peut toujours considérer ce modèle ou alors utiliser l'approche précédente par régression linéaire.

Prédiction des positions moyennes

Pour chaque observation du mot clé $\boldsymbol{i}$, l'approche par régression prédit toujours la même valeur. Ici pour une nouvelle observation, nous choisissons de tirer aléatoirement la valeur selon la loi normale tronquée avec les paramètres $\hat{\mu}_i$ et $\hat{\sigma_i^2}$. Le tirage consiste en un appel à un générateur aléatoire qui renvoie une réalisation de la variable aléatoire de loi normale tronquée choisie. Ainsi deux tirages consécutifs ne donnent pas la même valeur pour la position moyenne.

Après la position moyenne, nous devons également modéliser les impressions. Le modèle pour les impressions aura la position comme variable explicative. Ainsi si nos prédictions des positions sont constantes, celles des impressions le seront également. C'est ce qui motive notre choix de ne pas faire des prédictions constantes.

4.1.4 Comparaison des approches

Nous disposons de deux approches pour la variable position moyenne. Nous calculons l'erreur quadratique moyenne de prédiction sur des ensembles de tests en utilisant les deux approches. Pour rappel, l'erreur quadratique moyenne de prédiction se définit comme :

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{4.6}$$

où y_i et $\hat{y}_i$, $i = 1, \ldots, n$ sont respectivement les valeurs observées et les valeurs prédites de la position moyenne et n, la taille de l'ensemble de test. Sur tous nos exemples, l'approche par régression est celle qui donne l'erreur la plus faible. Ce qui tend à montrer que la première approche est meilleure. Or, ne pas prédire chaque observation exactement n'est pas très problématique. En réalité ce qui nous intéresse principalement c'est la prédiction du couple (position, impression). En effet c'est ce couple qui intervient principalement dans la prédiction du taux de clics et c'est sa bonne prédiction qui est plus importante. Par exemple, considérons deux observations $\mathbf{x}_0 = (2,5,\ 40)$ et $\mathbf{x}_1 = (4,\ 10)$ du couple (position, impression) et $\hat{\mathbf{x}}_0$ et $\hat{\mathbf{x}}_1$ leurs valeurs prédites respectives. Les prédictions idéales sont $\hat{\mathbf{x}}_0 = (2,5,\ 40)$ et $\hat{\mathbf{x}}_1 = (4,\ 10)$. Cependant nous considérons que les prédictions suivantes sont également bonnes $\hat{\mathbf{x}}_0 = (4,\ 10)$ et $\hat{\mathbf{x}}_1 = (2,5,\ 40)$. Dans ces dernières valeurs prédites, la valeur de la position pour $\mathbf{x}_0$ est fausse, puisque 4 est prédit au lieu de 2,5, mais l'impression prédite correspond à la valeur attendue ; pour une position 4, on s'attend à un nombre d'impressions de 10 et la valeur 10 est effectivement prédite. L'idée, c'est qu'étant donné la position moyenne prédite, il faut obtenir un nombre d'impressions conforme à cette position. Le plus important c'est que pour la position moyenne que nous prédisons (correcte ou non), nous ayons une impression prédite juste. C'est en cela que nous choisissons des prédictions aléatoires.

Néanmoins, nous devons nous assurer que nos positions prédites ne sont pas trop éloignées des valeurs réelles. Sur la figure 4.4, nous traçons les positions prédites en fonction des positions réelles : les valeurs sont rangées dans l'ordre croissant. Ce qui donne alors un graphique semblable à un diagramme quantile quantile. Nous l'avons fait pour différents mots clés. On note que les nuages de points présentent chacun une tendance linéaire. Ainsi, dans nos valeurs prédites, on retrouve des valeurs très proches des valeurs réelles.

Une autre métrique pour conforter nos prédictions consiste à calculer la médiane de l'erreur relative en valeur absolue. Nous calculons la quantité :

$$err_{med} = mediane\left(\{|y_j - \hat{y}_j|, j = 1, \ldots, n\}\right). \tag{4.7}$$

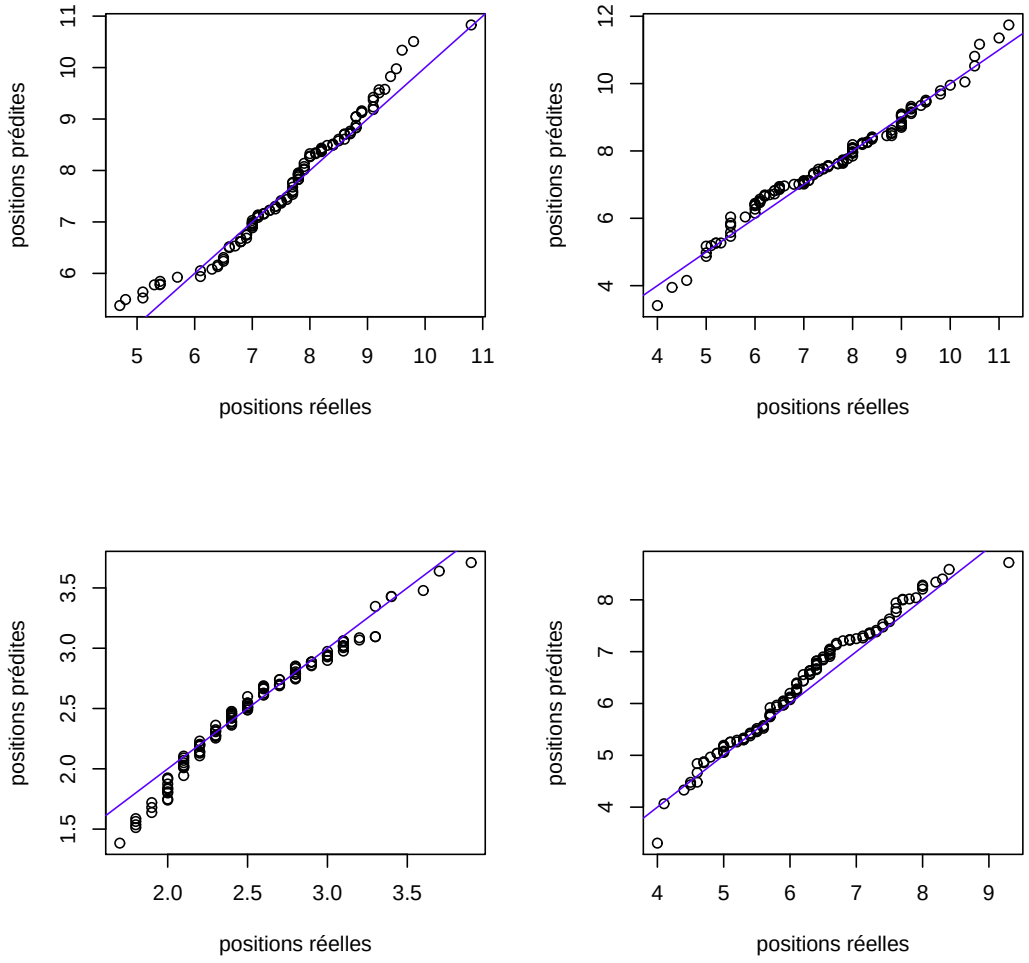

Figure 4.4 Nuage de points des positions prédites selon la loi normale tronquée en fonction des positions réelles pour quatre mots clés.

Pour une observation sur deux, l'erreur de prédiction commise est inférieure à la quantité err_{med}. La valeur optimale de cette métrique est 0 : dans ce cas plus de la moitié des observations sont correctement prédites. Nous sélectionnons 50 sous-ensembles de mots clés du jeu de données C : nous obtenons alors 50 nouveaux jeux de données plus petits. Sur chacun de ces derniers, nous calculons alors notre métrique. En moyenne nous trouvons que $err_{med} \approx 1,1$. Cette valeur est relativement petite ; ce qui veut dire qu'une observation sur deux est prédites avec une erreur inférieure ou égale à $1,1$.

La position maintenant modélisée, nous passons au nombre d'impressions.

4.2 Le nombre d'impressions

Pour rappel, le nombre d'impressions est le nombre de fois qu'une annonce textuelle donnée, associée à un mot clé, est apparue au cours d'une journée. On voit qu'il s'agit de compter le nombre de réalisations d'un évènement en l'occurrence l'évènement « l'annonce s'affiche ». Les impressions forment donc des données dites de « comptage ». La principale méthode pour analyser ce type de données est le modèle de Poisson. Toutefois ce dernier présente quelques limites. Ainsi d'autres méthodes ont été proposées ; celles-ci sont essentiellement des améliorations ou des modifications du modèle de Poisson. Pour nos données, nous avons exploré quelques unes de ces méthodes et nous avons retenu le modèle de type *hurdle* logistique et binomial négatif.

Dans la suite, nous présentons dans un premier temps les méthodes que nous avons considérées, puis nous les comparons entre elles pour en retenir la « meilleure » (pour nos données).

4.2.1 La régression de Poisson

Le modèle de régression de Poisson est très populaire dans l'analyse des données de comptage (Cameron et Trivedi, 2013). Par exemple le nombre de patients admis dans un hôpital, le nombre d'heures d'absence d'un étudiant au cours d'une session, etc. Tout comme la régression logistique, la régression de Poisson est un cas particulier des modèles linéaires généralisés. Comme son nom l'indique, la régression de Poisson repose sur la loi de Poisson. En effet, on suppose que la variable dépendante est distribuée selon une loi de Poisson.

Loi de Poisson

Une variable aléatoire X suit une loi de Poisson de paramètre $\lambda > 0$ si sa fonction de masse est donnée par

$$P(X = x) = p(x) = \begin{cases} \frac{\lambda^x}{x!} \exp(-\lambda) & \text{si } x \in \mathbb{N} \\ 0 & \text{sinon.} \end{cases} \tag{4.8}$$

La loi de Poisson possède la propriété particulière d'avoir une moyenne et une variance égales. En effet on a

$$\mathbb{E}(X) = \mathrm{Var}(X) = \lambda.$$

Le modèle

Considérons n observations indépendantes de la forme $\{(\mathbf{x}_i, y_i), i = 1, 2, \ldots, n\}$ où les y_i sont les réalisations de la variable dépendante Y et chaque $\mathbf{x}_i$ est une réalisation du vecteur aléatoire $X = (X_1, X_2, \ldots, X_k)^T$ de variables explicatives. Dans un modèle de Poisson, on suppose que Y est distribuée selon une loi de Poisson conditionnellement au vecteur de covariables $\mathbf{x}_i$ soit $Y|X = \mathbf{x}_i \sim \mathcal{P}(\mu_i)$. Comme d'habitude, nous cherchons à modéliser la moyenne $\mu_i = \mathbb{E}(Y|X = \mathbf{x}_i)$. Toutes nos observations étant positives ou nulles, cette moyenne est donc strictement positive. Ainsi le modèle linéaire n'est pas envisageable puisqu'il autorise des valeurs négatives aussi. À l'image de la fonction *logit* pour la régression logistique, nous appliquons une transformation de la moyenne ici aussi : la transformation logarithmique. Ainsi on ajuste un modèle linéaire au logarithme de la moyenne. Le modèle de la régression de Poisson s'écrit alors :

$$\log(\mu_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_k x_{ik},\ i = 1, 2, \ldots, n, \tag{4.9}$$

où $\beta_0, \beta_1, \ldots, \beta_k$ sont des paramètres réels.

En prenant l'exponentielle à droite et gauche dans l'équation (4.9), on retrouve la moyenne

$$\begin{aligned} \mu_i &= \exp(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_k x_{ik}) \\ &= \exp(\mathbf{x}_i^T \beta),\ i = 1, 2, \ldots, n. \end{aligned}$$

En raison de la transformation logarithmique de la moyenne, le modèle de Poisson est aussi appelé modèle log linéaire.

Estimation des paramètres

Dans le modèle (4.9), nous devons estimer les paramètres $\beta = (\beta_0, \beta_1, \ldots, \beta_k)^T$. L'estimation se fait en utilisant la méthode de vraisemblance maximale. Nous savons que chaque $Y|\mathbf{x}_i$ suit une loi de Poisson de paramètre μ_i. Ainsi sous l'hypothèse d'indépendance des observations et en considérant notre échantillon précédent, la vraisemblance s'écrit

$$\mathcal{L}(\beta; y_1, y_2, \ldots, y_n) = \prod_{i=1}^{n} \frac{\mu_i^{y_i}}{y_i!} \exp(-\mu_i). \tag{4.10}$$

La log vraisemblance s'obtient directement

$$\begin{aligned}\ell(\beta; y_1, y_2, \ldots, y_n) &= \ln \mathcal{L}(\beta; y_1, y_2, \ldots, y_n) \\ &= \sum_{i=1}^{n} \left(y_i \, \mathbf{x}_i^T \beta - \exp(\mathbf{x}_i^T \beta) - \ln(y_i!) \right).\end{aligned}$$

Le troisième terme de la somme ne dépend pas de β et peut être abandonné. Ainsi l'estimateur de vraisemblance maximale $\hat{\beta}$ est tel que

$$\hat{\beta} = \arg\max_{\beta} \sum_{i=1}^{n} \left(y_i \, \mathbf{x}_i^T \beta - \exp(\mathbf{x}_i^T \beta) \right). \tag{4.11}$$

Pour obtenir $\hat{\beta}$, le système d'équations suivant est résolu :

$$\frac{\partial \ell(\beta; y_1, y_2, \ldots, y_n)}{\partial \beta} = 0 \iff \sum_{i=1}^{n} \mathbf{x}_i \left(y_i - \exp(\mathbf{x}_i^T \beta) \right) = \mathbf{0}. \tag{4.12}$$

On obtient un système de $k+1$ équations non linéaires. Ce système n'admet pas de solution analytique. Tout comme pour la régression logistique, l'algorithme de Newton-Raphson est généralement utilisé pour calculer $\hat{\beta}$.

Limites de la régression de Poisson

Le modèle de régression de Poisson présente quelques limites dans l'analyse des données de comptage. La principale limitation de ce modèle est liée à la propriété d'égalité de la moyenne et de la variance pour une loi de Poisson. En effet dans beaucoup de données, cette propriété d'équi-dispersion n'est pas respectée ; on a souvent affaire à des données avec une variance plus grande que la moyenne. On parle alors de sur-dispersion des données. Dans de rares cas, on peut avoir une sous-dispersion.

Un autre problème fréquent et traité dans la littérature est l'excès de "zéro" dans les données. Il arrive que les données contiennent un nombre de "zéro" très largement supérieur à celui attendu dans le cas d'une loi de Poisson. Notons que dans notre cas, nous avons plutôt éventuellement un excès de "un" car le nombre d'impressions est strictement positif. Néanmoins comme nous le montrons plus bas, nous pouvons nous ramener à un cas avec excès de "zéro" et donc appliquer les méthodes qui traitent ce problème.

Pour répondre à ces deux problématiques, beaucoup de modifications et de généralisations du modèle de Poisson ont été proposées. Ici nous en présentons quelques unes. D'abord la régression quasi-Poisson et la régression binomiale négative permettent de considérer la sur-

dispersion dans les données. Ensuite nous présentons les modèles à "obstacle" (*hurdle models*) et les modèles à "excès de zéros" (*zero inflated models*) pour traiter le grand nombre de zéros.

4.2.2 Sur-dispersion dans les données

Dans cette section, nous présentons deux méthodes permettant de modéliser la sur-dispersion dans les données, c'est-à-dire le cas où la variance est supérieure à la moyenne. La première méthode est une modification du modèle de Poisson tandis que la seconde fait l'hypothèse que la variable dépendante suit une loi binomiale négative plutôt qu'une loi de Poisson. Pour introduire ces deux modèles, une brève présentation du modèle linéaire généralisé est nécessaire.

Le modèle linéaire généralisé

Dans le modèle de régression linéaire classique, l'une des principales hypothèses est la normalité de la variable de réponse Y. Le modèle linéaire généralisé permet de considérer une réponse Y qui est distribuée selon d'autres lois. Néanmoins sa distribution doit appartenir à la famille exponentielle. Ensuite selon cette distribution, une fonction dite *lien* permet alors d'associer la moyenne de Y à un prédicteur linéaire c'est-à-dire que

$$g(\mu) = X^T\beta,$$

où β est le vecteur des paramètres et $X = (1, X_1, X_2, \ldots, X_k)^T$ le vecteur des variables explicatives. Par exemple, dans le cas de la régression logistique binaire, Y suit une loi de Bernoulli avec la fonction *logit* comme fonction *lien*. Pour la régression de Poisson, Y suit une loi de Poisson avec le logarithme comme fonction *lien*. Pour un membre de la famille exponentielle, la fonction de densité de la variable Y étant donné le vecteur des variables explicatives $\mathbf{x}$ possède la forme générale (Cameron et Trivedi, 2013; Charpentier, 2013)

$$f(y|\theta, \phi) = \exp\left(\frac{y\,\theta - b(\theta)}{a(\phi)} + c(y, \phi)\right), \tag{4.13}$$

où θ est le paramètre d'intérêt ; ϕ est un paramètre de dispersion ou de nuisance généralement connu. Les fonctions $a(.)$ et $b(.)$ sont telles que

$$\mathbb{E}(Y) = \mu = b^{'}(\theta) \quad \text{et} \quad \text{Var}(Y) = a(\phi)b^{''}(\theta). \tag{4.14}$$

Elles sont caractéristiques de chaque loi. Par exemple pour une loi de Bernoulli de paramètre p, on a $\theta = \log\left(\frac{p}{1-p}\right)$, $a(\phi) = 1$, $b(\theta) = \log(1 + \exp(\theta))$ et $c(y, \phi) = 0$.

La fonction *lien* particulière qui associe la moyenne μ au paramètre θ est appelée fonction *lien canonique*. En effet différentes fonctions *lien* sont possibles pour certains modèles.

L'estimation des paramètres se fait par la méthode de vraisemblance maximale. On pose $\mu_i = \mathbb{E}(Y|\mathbf{x}_i)$ et $g(\mu_i) = \mathbf{x}_i^T\beta$. Il nous suffit de calculer le vecteur gradient et de l'annuler. Pour une observation $(\mathbf{x}_i, y_i)$ donnée, la log vraisemblance est :

$$\ell(\beta; y_i) = \ln(f(y_i|\theta_i, \phi)) = \frac{y_i\,\theta_i - b(\theta_i)}{a(\phi)} + c(y, \phi).$$

Pour un β_j quelconque, $j = 1, 2, \ldots, k$, on a alors :

$$\frac{\partial \ell(\beta; y_i)}{\partial \beta_j} = \frac{\partial \ell(\beta; y_i)}{\partial \theta_i}\frac{\partial \theta_i}{\partial \mu_i}\frac{\partial \mu_i}{\partial \beta_j} = \frac{y_i - b'(\theta_i)}{a(\phi)} \times \frac{1}{b''(\theta_i)} \times \frac{\partial \mu_i}{\partial \beta_j}$$

En utilisant les équations (4.14) et toutes les observations, l'estimateur de maximum de vraisemblance $\hat{\beta}$ est alors solution du système d'équations (McCullagh et Nelder, 1989) :

$$\sum_{i=1}^{n} \frac{y_i - \mu_i}{\mathrm{Var}(Y_i)}\frac{\partial \mu_i}{\partial \beta_j} = 0 \quad \forall j = 1, 2, \ldots, k. \tag{4.15}$$

L'algorithme de Newton-Raphson ou l'algorithme des moindres carrés pondérés permettent d'obtenir l'estimateur de maximum de vraisemblance $\hat{\beta}$.

Le modèle quasi-Poisson

La loi de Poisson appartient à la famille exponentielle avec $\theta = \ln(\lambda)$, $b(\theta) = \exp(\theta)$, $c(y, \phi) = -\ln(y!)$ et $a(\phi) = 1$. En appliquant les équations (4.14), on retrouve bien l'égalité de la moyenne et de la variance c'est-à-dire $\mathbb{E}(Y|\mathbf{x}) = \mathrm{Var}(Y|\mathbf{x}) = \exp(\theta)$. Pour briser cette égalité, on conserve le même paramètre θ ainsi que les fonctions $b(.)$ et $c(.)$ de la loi de Poisson mais on ne fixe plus le paramètre de dispersion. On définit alors $a(\phi) = \phi > 1$. Ainsi on a $\mathrm{Var}(Y|\mathbf{x}) = \phi \exp(\theta) = \phi\,\mathbb{E}(Y|\mathbf{x})$. On parle alors de loi quasi-Poisson. Notons toutefois qu'il ne s'agit pas d'une loi de probabilité.

Le paramètre ϕ permet maintenant de tenir compte de l'effet de sur-dispersion. L'équation du modèle quasi-Poisson est la même que celle du modèle de Poisson (4.9). En fait le modèle de Poisson et le modèle quasi-Poisson donnent les mêmes coefficients $\hat{\beta}$. La principale différence réside dans l'inférence sur les paramètres. En effet, les matrices de covariances des estimateurs sont différentes.

Une estimation du paramètre de dispersion ϕ se base sur les résidus de Pearson (Cameron

et Trivedi, 2013). On obtient alors l'estimateur du χ^2 de Pearson $\hat{\phi}$ défini par (Cameron et Trivedi, 2013)

$$\hat{\phi} = \frac{1}{n-k} \sum_{i=1}^{n} \frac{(y_i - \hat{\mu}_i)^2}{\hat{\mu}_i}, \tag{4.16}$$

où n le nombre d'observations et k le nombre de variables explicatives dans le modèle. On peut noter que si on définit le paramètre ϕ inférieur à 1 alors on peut gérer les rares cas de sous-dispersion dans les données.

Le modèle binomial négatif

La régression quasi-Poisson est une première alternative pour traiter l'effet de sur-dispersion. L'idée ici consiste à choisir une autre loi pour la distribution de la variable Y en l'occurrence une loi dont l'espérance et la variance ne sont pas toujours égales. C'est ainsi que la loi binomiale négative est proposée.

La loi binomiale négative

Il existe différentes façons de définir la loi binomiale négative. Elle apparaît par exemple comme le nombre d'échecs nécessaires à la réalisation d'un nombre de succès donné dans une série d'épreuves de Bernoulli indépendantes. L'approche qui nous intéresse consiste à voir la loi binomiale négative comme un mélange de loi de Poisson. En effet on suppose que les données suivent une loi de Poisson (équation 4.8) de paramètre λ mais avec λ lui même aléatoire et distribué selon une loi de Gamma. Une variable aléatoire X est dite de loi gamma de paramètres $r > 0$ et $\theta > 0$ si et seulement si elle est à valeurs dans $\mathbb{R}_+^*$ et sa fonction de densité est :

$$f(x) = \frac{\theta^r}{\Gamma(r)} e^{-\theta x} x^{r-1}, \tag{4.17}$$

où r est un paramètre de forme et θ un paramètre d'intensité.
On a donc $\lambda \sim \text{Gamma}(r, \theta)$ et pour λ fixé, $Y|\lambda \sim \text{Poisson}(\lambda)$. La variable Y suit alors une loi binomiale négative de paramètres r et avec $p = \frac{1}{1+\theta}$. La fonction de masse d'une variable aléatoire Y de loi binomiale négative est la fonction de densité marginale du vecteur aléatoire $[Y, \lambda]$. Ainsi la fonction de masse s'obtient :

$$\begin{aligned} p(y; r, \theta) &= \int_0^\infty p(y|\lambda) \times f(\lambda; r, \theta) \, \mathrm{d}\lambda \\ &= \int_0^\infty \exp(-\lambda) \frac{\lambda^y}{y!} \times \frac{\lambda^{r-1} \theta^r \exp(-\theta\lambda)}{\Gamma(r)} \, \mathrm{d}\lambda \\ &= \frac{\theta^r}{\Gamma(r) y!} \int_0^\infty \lambda^{y+r-1} \exp(-\lambda(\theta+1)) \, \mathrm{d}\lambda. \end{aligned}$$

Le changement de variable $u = \lambda(\theta + 1)$ conduit à :

$$\begin{aligned} p(y; r, \theta) &= \frac{\theta^r}{\Gamma(r) y! (\theta+1)^{y+r}} \int_0^\infty u^{y+r-1} \exp(-u)\, \mathrm{d}u \\ &= \frac{\Gamma(r+y)}{\Gamma(r) y!} \left(\frac{1}{1+\theta}\right)^y \left(\frac{\theta}{1+\theta}\right)^r, \; y \in \mathbb{N}, \end{aligned}$$

où $\Gamma(.)$ est la fonction Gamma d'Euler et r et θ sont les paramètres de la loi Gamma suivie par λ. Une écriture alternative de la fonction de masse utilisant l'espérance $\mu = \mathbb{E}(Y) = \frac{r}{\theta}$ est

$$p(y; \mu, r) = \frac{\Gamma(r+y)}{\Gamma(r) y!} \frac{r^r \mu^y}{(r+\mu)^{r+y}}, \; y \in \mathbb{N}. \tag{4.18}$$

L'espérance et la variance d'une loi binomiale négative sont

$$\mathbb{E}(Y|\mu, r) = \mu \quad \text{et} \quad \mathrm{Var}(Y|\mu, r) = \mu + \frac{\mu^2}{r} = \mu + \alpha\mu^2 \quad \text{où } \alpha = \frac{1}{r}. \tag{4.19}$$

On note ici que la moyenne et la variance ne sont jamais égales ; on remarque surtout que la variance est toujours supérieure à la moyenne. L'une des différences entre la régression quasi-Poisson et la régression binomiale négative est la forme de l'expression de la variance. Dans le premier modèle la relation entre la variance et la moyenne est linéaire ($\mathrm{Var}(Y) = \phi\mu$) tandis que dans le second, elle est quadratique.

Le modèle

Considérons une réponse dépendante Y et k variables indépendantes ou explicatives X_1, X_2, …, X_k. Pour une réalisation $\mathbf{x}_i$ du vecteur des variables indépendantes, la variable $Y|\mathbf{x}_i$ suit une loi binomiale négative de moyenne $\mu_i = \mathbb{E}(Y|\mathbf{x}_i)$ et avec un paramètre α tel que donné par l'équation (4.19). Le modèle de régression binomiale négative est aussi un modèle log linéaire. Il s'écrit

$$\log(\mu) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k, \tag{4.20}$$

où β_0, $\beta_1, \ldots, \beta_k$ sont des paramètres réels.

Estimation des paramètres

Si le paramètre α est connu alors la régression binomiale négative entre dans le cadre des modèles linéaires généralisés. Ainsi l'estimation des paramètres se fait par la méthode de maximum de vraisemblance en utilisant les équations (4.15). Par exemple on peut noter que pour $\alpha = 1$, la loi binomiale négative devient une loi géométrique.

Par contre si le paramètre α doit être estimé, il existe deux possibilités. La première consiste

à utiliser une estimation $\hat{\alpha}$ dans les équations (4.15) et utiliser la procédure d'optimisation des modèles linéaires généralisés. Un estimateur convergent (Cameron et Trivedi, 2013) de α est

$$\hat{\alpha} = \frac{1}{n-k} \sum_{i=1}^{n} \frac{(y_i - \hat{\mu}_i)^2 - \hat{\mu}_i}{\hat{\mu}_i^2}.$$

Dans la deuxième méthode, les paramètres β et α sont estimés simultanément par la méthode de vraisemblance maximale.

Nous disposons maintenant de deux méthodes pour traiter la sur-dispersion dans les données. Comme dit plus haut, un autre problème dans les données de comptage est l'excès de "zéros". La section suivante présente donc deux catégories de méthodes permettant de traiter ce phénomène.

4.2.3 Les modèles de type *hurdle* et *zero-inflated*

Les modèles de type *hurdle* et *zero-inflated* permettent de considérer l'excès de valeurs nulles dans les données. Ce phénomène survient lorsque la proportion de "zéros" dans les données est largement supérieure à celle attendue pour une loi de Poisson ou une loi binomiale négative. Un exemple est le nombre de jours d'hospitalisation des patients dans un hôpital : beaucoup de personnes vont à l'hôpital sans pour autant y être hospitalisées générant ainsi un grand nombre d'observations nulles. Zuur et al. (2009) proposent un exemple de données avec excès de "zéros" sur le comptage des hippopotames dans un milieu.

Le modèle de type *hurdle* est un modèle en deux parties tandis que le modèle de type *zero-inflated* est plutôt un mélange de modèles. La principale différence entre ces deux types de modèles réside dans la modélisation des zéros. En effet dans le modèle *hurdle*, tous les "zéros" sont issus d'un même processus alors que le modèle *zero-inflated* suppose deux sources pour les "zéros". La figure 4.5 illustre cette distinction sur l'exemple des hippopotames de (Zuur et al., 2009) d'où les images sont également tirées. Zuur et al. (2009) comptent le nombre d'hippopotames dans un milieu. Sur l'image de gauche, dans le cas du modèle *hurdle*, on ne distingue pas les raisons de l'absence d'un compte nul tandis qu'à droite, pour le cas *zero-inflated*, l'absence d'un hippopotame se justifie par des motifs différents.

Les modèles de type *hurdle*

Dans ce modèle aucune distinction n'est faite sur les zéros : tous proviennent d'un même processus. Il s'agit d'un modèle en deux parties :

1. un premier modèle (généralement la régression logistique) détermine d'abord la proba-

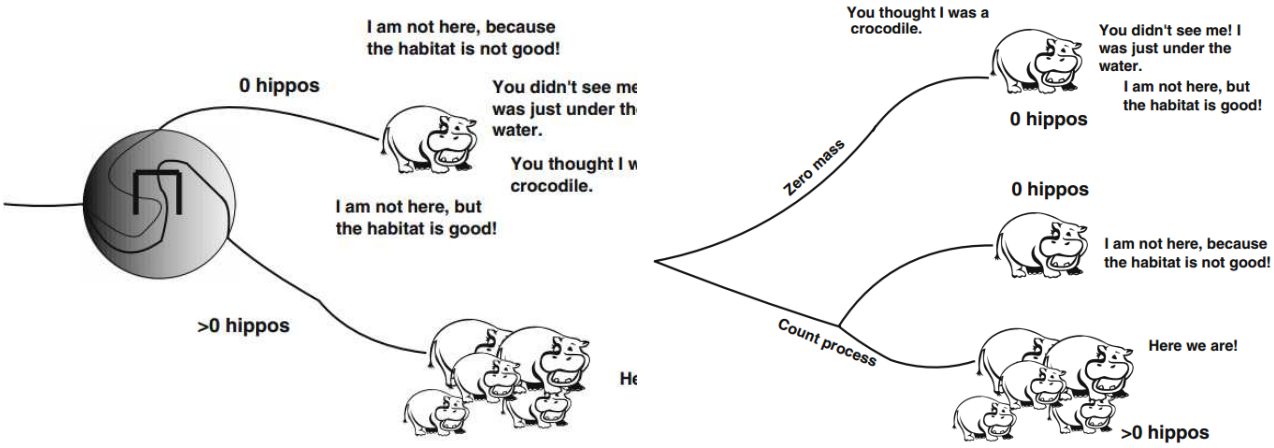

Figure 4.5 Illustration de la modélisation des "zéros" sur un exemple dans les modèles de type *hurdle* à gauche et "zero-inflated" à droite.

bilité qu'une observation soit nulle ou non.

2. Ensuite un modèle de comptage tronqué comme Poisson ou binomial négatif est alors ajusté pour les données strictement positives. Les lois (Poisson ou binomiale négative) sont tronquées afin de ne pas générer de zéros.

Ce modèle tire son nom de ce mécanisme. En effet pour une observation donnée, une barrière (*hurdle* en anglais) doit d'abord être franchie (celle-ci doit être classée comme strictement positive) avant d'être ensuite prédite plus finement.

Écriture du modèle

Considérons une variable de réponse Y et k variables explicatives X_1, $X_2, \ldots, X_k$. Dans un modèle de type *hurdle*, la fonction de masse de la variable $Y|\mathbf{x}$ s'écrit :

$$p_{hurdle}(y;\beta,\gamma) = \begin{cases} p_1(0;\gamma) = P(Y=0|\mathbf{x}) & \text{si } y = 0 \\ \frac{1-p_1(0;\gamma)}{1-p_2(0;\beta)} p_2(y;\beta) & \text{si } y > 0. \end{cases} \tag{4.21}$$

La fonction de masse $p_1(.)$ est généralement celle d'une loi de Bernoulli. Toutefois, toute autre loi tronquée à droite à 1 et positive est également envisageable. Nous considérons uniquement le cas d'une loi Bernoulli car celle-ci conduit au modèle logistique binaire qui donne des probabilités qui s'interprètent bien.
La fonction de masse $p_2.()$ est celle d'une loi tronquée à gauche à 1. Nous considérons la loi de Poisson tronquée ainsi que la loi binomiale négative tronquée.
Le vecteur des paramètres du premier modèle γ contient les coefficients associés aux variables explicatives considérées dans ce modèle. En effet on peut choisir toutes les variables explica-

tives ou uniquement un sous-ensemble qu'on note Z.
Le vecteur des coefficients de régression du modèle de comptage est β. Ici on prend généralement le vecteur X des variables indépendantes. Néanmoins on peut aussi garder uniquement un sous-ensemble.

Notons $p_i = P(Y > 0|\mathbf{x}_i)$ la probabilité que Y soit non nulle sachant $\mathbf{x}_i$, $\mu_i^{'}$ la moyenne de la loi de comptage (Poisson ou binomiale négative) non tronquée et
$\mu_i = \mathbb{E}(Y|\mathbf{x}_i) = \frac{1-p_1(0;\gamma)}{1-p_2(0;\beta)}\mu_i^{'}$ la moyenne. Le modèle *hurdle* s'écrit alors :

$$\begin{cases} \ln\left(\frac{p_i}{1-p_i}\right) = & z_i^T\gamma \\ \ln(\mu_i) = & \mathbf{x}_i^T\beta + \ln(1-p_1(0;\gamma)) - \ln(1-p_2(0;\beta)) \end{cases} \tag{4.22}$$

Estimation des paramètres
L'estimation des paramètres γ et β se fait par la méthode de maximum de vraisemblance. Ici la log vraisemblance des observations se décompose en deux parties :

$$\ell(\beta, \gamma\,;\, y_1, \ldots, y_n) = \ell_1(\gamma\,;\, y_1, \ldots, y_n) + \ell_2(\beta\,;\, y_1, \ldots, y_n) \tag{4.23}$$

avec

$$\ell_1(\gamma; y_1, \ldots, y_n) = \sum_{i=1}^{n} \mathbb{1}(y_i = 0) \ln p_1(0;\gamma) + \sum_{i=1}^{n}(1 - \mathbb{1}(y_i = 0)) \ln p_1(1;\gamma) \tag{4.24}$$

et

$$\ell_2(\beta\,;\, y_1, \ldots, y_n) = \sum_{i=1}^{n}(1 - \mathbb{1}(y_i = 0)) \ln p_2(y_i\,;\, \beta). \tag{4.25}$$

On peut donc maximiser séparément les deux quantités. Les techniques pour maximiser ℓ_1 et ℓ_2 ont déjà été présentées dans ce mémoire.

Le modèle de type *hurdle* est une famille de modèles. Pour la suite, nous considérons les deux modèles suivants :

- HP : Bernoulli avec Poisson ; nous choisissons un modèle logistique pour la première partie et le modèle poissonien pour le second.
- HBN : Bernoulli avec binomiale négative.

Les modèles de type *zero-inflated*

Dans les modèles *zero-inflated*, deux processus génèrent les observations nulles : un premier processus qui génère toujours des "zéros" et un second (processus de comptage) qui génère des valeurs positives ou nulles. Ainsi une observation nulle a une probabilité p de provenir

du processus "nul" et une probabilité $1-p$ d'être issue du processus de comptage. Le modèle apparaît alors comme un mélange d'une fonction de masse concentrée en 0 et une distribution de comptage (Poisson ou binomiale négative).

Soient $\mathbf{x}_i$ une réalisation du vecteur des variables explicatives et Y la variable de réponse. On a :

$$\begin{aligned} Y|\mathbf{x}_i = 0 \quad & \text{avec une probabilité } p_i \\ Y|\mathbf{x}_i \sim \mathcal{D}(\mu_i) \quad & \text{avec une probabilité } (1-p_i), \end{aligned}$$

où $\mathcal{D}$ désigne la loi de Poisson ou la loi binomiale négative. Lambert (1992), la première à proposer le modèle *zero-inflated*, suggère d'ajuster un modèle logistique pour modéliser la probabilité p_i. Comme pour le modèle *hurdle*, l'ensemble ou un sous-ensemble Z des variables explicatives est utilisé. Les régressions Poisson ou binomiale négative (vues plus haut) sont choisies pour modéliser la deuxième partie. Notons ici que p_i est la probabilité que $Y|\mathbf{x}_i$ soit nulle. Même si le choix de la régression logistique semble évident (à cause de son interprétation comme probabilité), d'autres méthodes peuvent être considérées.

Il faut estimer les paramètres γ du modèle logistique et β du modèle de comptage. On notera que dans le cas de la régression binomiale négative, un paramètre supplémentaire à savoir le paramètre de dispersion doit également être estimé. La méthode de maximum de vraisemblance est bien sûr utilisée. Contrairement au modèle *hurdle*, la log vraisemblance ne se décompose pas en deux parties indépendantes. L'optimisation est alors légèrement plus compliquée. Lambert (1992) propose l'utilisation de l'algorithme EM ou d'espérance maximisation (Wu, 1983) pour maximiser la vraisemblance. Toutefois l'algorithme de Newton-Raphson donne des résultats satisfaisants. Dans les modèles *zero-inflated*, nous considérons le modèle ZIP pour logistique + modèle de Poisson et le modèle ZIBN pour logistique + modèle binomial négatif.

4.2.4 Comparaison des modèles

Nous avons présenté sept modèles :

- modèle de Poisson
- modèle quasi-Poisson
- modèle binomial négatif
- modèle *hurdle* Poisson (HP)
- modèle *hurdle* binomial négatif (HBN)
- modèle *zero-inflated* Poisson (ZIP)
- modèle zero-inflated binomial négatif (ZIBN)

Nous devons maintenant les comparer. Une première méthode consiste d'abord à faire une validation de chaque modèle par une analyse des résidus par exemple. On écarte ensuite

les modèles qui semblent incorrects. La technique de comparaison la plus simple consiste à utiliser les critères d'information tels que l'AIC (*Akaike Information Criterion*) ou le BIC (*Bayesian Information Criterion*). Pour un modèle donné, ces critères sont définis comme suit (Aho et al., 2014) :

$$\text{AIC} = -2 \ln \hat{\mathcal{L}} + 2p, \tag{4.26}$$

$$\text{BIC} = -2 \ln \hat{\mathcal{L}} + p \ln n, \tag{4.27}$$

où $\hat{\mathcal{L}}$ est la fonction de vraisemblance estimée du modèle, p le nombre de paramètres estimés dans le modèle et n la taille de l'échantillon.
Le meilleur modèle est celui qui minimise ces critères.

Présentation de nos expériences

Les impressions dont nous disposons sont supérieures ou égales à un. Afin d'utiliser les méthodes avec excès de "zéros", nous soustrayons une unité à toutes nos impressions et nous utilisons la relation

$$\mathbb{E}(X + 1) = \mathbb{E}(X) + 1.$$

En effet nos modèles prédisent des moyennes à la fin. Ainsi nous retrouvons les prédictions pour les données initiales en rajoutant 1.
Nous avons constaté que dans tous nos jeux de données, nous avons un phénomène de sur-dispersion puisque la variance des observations est largement supérieure à la moyenne ; ce qui indique que le modèle de Poisson risque de ne pas être très bon comme nous le confirmons plus bas. Quant à l'excès de "zéros" (excès de "un" en réalité), toutes les données ne semblent pas présenter ce phénomène.

Nous disposons de deux variables explicatives : la position moyenne (*AveragePosition*) et les mots clés (*Keyword_id*). Nous utilisons les deux variables dans tous nos modèles. Dans les modèles *hurdle* et *zero-inflated*, les deux variables sont utilisées pour la régression logistique et pour le modèle de comptage. Toutefois nous avons dû quelques fois retirer la variable *Keyword_id* du modèle de comptage pour le modèle *zero-inflated* en raison de contraintes numériques entraînant la non convergence des algorithmes d'optimisation.

Le tableau 4.2 présente les valeurs de l'AIC et du BIC obtenues pour chacune des méthodes sur le jeu de données A. Il n'y pas de vraisemblance dans le cas du modèle quasi-Poisson car la "loi" quasi-Poisson n'est pas une loi de probabilité ; nous ne pouvons donc pas calculer ces quantités. Toutefois nous avons une estimation du coefficient de dispersion ϕ qui nous donne une idée sur la dispersion.
Le code permettant d'obtenir le tableau 4.2 est disponible à l'annexe B.

Tableau 4.2 Tableau comparatif des méthodes pour le jeu de données A.

	Poisson	BN	Quasi-Poisson	ZIBN	ZIP	HBN	HP
Df	34	35	36	37	67	69	68
AIC	51153,80	28981,69	NA	28873,54	63647,19	28840,29	51096,71
BIC	51365,63	29199,75	NA	29104,06	64064,62	29270,18	51520,37

La ligne Df représente le nombre de paramètres estimés dans chaque modèle. Le coefficient de dispersion obtenu dans ce cas est $\hat{\phi} = 9,31$; ce qui est assez élevé. Pour rappel pour une loi de Poisson, ce coefficient doit être égal à 1. Ce qui veut dire que les données sont très dispersées. Ainsi tous les modèles avec la binomiale négative ont un AIC et un BIC plus faible. Selon le critère de l'AIC, le meilleur modèle est le modèle HBN tandis que c'est le modèle ZIBN qui l'emporte selon le BIC. Ce jeu de données ne semble pas présenter un excès de "zéros" car la régression binomiale négative donne un AIC et un BIC très proche des minimums obtenus.

Du jeu de données C, nous avons créé une vingtaine de jeux de données et comparer les sept méthodes. Sur tous les cas testés, le modèle HBN est celui qui donne la plus petite valeur pour l'AIC et le BIC. Nous avons donc choisi de retenir ce modèle pour la suite. Normalement pour chaque nouveau jeu de données, une comparaison des méthodes est nécessaire. Néanmoins, nous souhaitons avoir un modèle général ; nous retenons donc le modèle de type *hurdle* Bernoulli avec binomiale négative.

Utilisation du modèle

Normalement une validation du modèle serait nécessaire : une analyse des résidus par exemple, des tests individuels sur les variables, un test global. Cette analyse est très similaire à celle présentée dans le cadre de la régression logistique avec les tests de Wald notamment. Nous avons rajouté un terme quadratique dans la régression binomiale négative. En effet le carré de la position moyenne s'est avérée être une variable significative ; dans le test de Wald (équation 3.15), l'hypothèse H_0 est rejetée.

Notre modèle prédit les moyennes $\hat{\mu}_i$ qui sont des valeurs continues. Or nous souhaitons des prédictions entières. Nous considérons alors la partie entière afin d'avoir des prédictions entières et nous rajoutons l'unité que nous avions soustraite initialement.

Pour valider nos prédictions, une première métrique que nous considérons est l'erreur quadratique moyenne de prédiction (MSE). Nous la comparons au modèle simple qui prédit toutes les observations par la moyenne des données. L'erreur obtenue avec notre modèle n'est pas

toujours plus petite que celle du modèle simple. En effet le MSE est très sensible ; il peut être biaisé par quelques observations très mal prédites. Par exemple si nous avons une observation qui vaut 500 et que nous prédisons 400, cela fait une différence de 100 qui une fois au carrée fait exploser l'erreur. Nous avons plutôt préféré calculer des pourcentages de bonne classification avec des marges autorisées, c'est-à-dire qu'on autorise une erreur de quelques unités sur la valeur prédite. Nous créons 50 jeux de données constitués de 30 mots clés à partir du jeu de données C ; nous ajustons notre modèle sur un ensemble d'apprentissage et faisons des prédictions sur l'ensemble de test. Les résultats sont présentés dans la deuxième colonne du tableau 4.3 (Données 1). Nous créons ensuite 50 autres jeux de données telles que les nombres d'impressions soient inférieurs à 100 (Données 2) pour tous les mots clés.

Tableau 4.3 Évolution du pourcentage de bonne prédiction en fonction des marges d'erreurs permises.

$\lvert y_i - \hat{y}_i \rvert$	Données 1	Données 2 (Impressions ≤ 100)
0	0,18	0,21
≤ 1	0,37	0,41
≤ 2	0,48	0,53
≤ 5	0,65	0,69
≤ 10	0,75	0,8

Pour les données 1, on constate qu'environ une observation sur deux est prédite avec une erreur d'au plus deux impressions. Aussi si on autorise une erreur d'au plus 10 impressions, 75% des observations sont alors bien prédites. Les impressions prennent des valeurs allant de 1 à 3000 ; il est évident qu'une erreur de 100 pour une observation de 2500 n'est pas énorme. En considérant les mots clés tels que le nombre d'impressions soit inférieur ou égal à 100 (Données 2), on se rend compte que les prédictions sont meilleures puisque par exemple environ 70% des données sont bien prédites avec une marge de 5.
En somme nous considérons les prédictions de notre modèle satisfaisantes. Le nombre d'impressions à son tour est utilisé comme variable explicative pour modéliser la variable coût.

4.3 La variable coût

Le coût est la dernière variable inconnue que nous utilisons dans notre modèle global de prédiction des taux de clics. Cette variable représente le montant quotidien total que l'annonceur

a payé pour l'affichage d'une annonce. Nous avons des données issues d'une tarification au coût par clic (CPC) c'est-à-dire que l'annonceur ne paie uniquement que si un utilisateur clique sur son annonce. On peut déjà voir qu'une annonce qui ne génère pas de clics aura nécessairement un coût nul. Il y a donc une relation assez forte entre les coûts et les taux de clics. En réalité, tout ce projet aurait pu porter sur la modélisation des coûts tant cette variable est également importante pour un annonceur. Néanmoins une exploration des données montre que pour les coûts strictement positifs c'est-à-dire les observations à clics non nuls, la relation entre ces deux variables est moins immédiate. On note notamment, pour le jeu de données A, un coefficient de corrélation linéaire pas très élevé de l'ordre de $0,36$ entre le taux de clics et le coût. De plus sur des jeux de données extraits du jeu de données C, on se rend compte que la corrélation est parfois positive parfois négative, montrant ainsi la complexité de la relation entre ces deux variables.

La variable coût présente deux caractéristiques dont il faut tenir compte dans sa modélisation. D'une part il s'agit de données positives ou nulles. D'autre part, comme pour le taux de clics, un fort pourcentage des valeurs est nul. D'abord à cause de la positivité des données, le modèle linéaire est exclu. Ensuite, il n'existe quasiment pas de méthodes permettant de prédire exactement des valeurs nulles alors que nous en avons beaucoup.

Notre idée consiste alors à utiliser un modèle de type *hurdle* comme pour les impressions. On suppose que toutes les valeurs nulles proviennent d'un processus qui ne génère que des 0 et les valeurs non nulles quant à elles sont issues d'un processus qui génère des valeurs strictement positives. Nous avons donc un modèle en deux parties : le modèle logistique pour la première et un modèle "strictement positif" pour la seconde.

On se réfèrera au chapitre 1 pour la régression logistique. Quant au modèle "strictement positif", deux méthodes sont proposées : une première basée sur la transformation des données et un modèle linéaire généralisé, le modèle Gamma.

4.3.1 Le modèle lognormal

Pour modéliser les valeurs strictement positives, une transformation logarithmique des données est proposée. De plus nous supposons que la variable coût suit une loi lognormale pour les données non nulles.

Une variable aléatoire continue est dite de loi lognormale si le logarithme de cette variable suit une loi normale. Précisément, X est dite de loi lognormale de paramètres μ_Y et σ_Y^2 si la variable $Y = \ln X$ suit une loi normale $\mathcal{N}(\mu_Y, \sigma_Y^2)$.

Quatre jeux de données sont extraits du jeu de données C en tirant aléatoirement des mots

clés. Nous traçons l'histogramme du logarithme des coûts strictement positifs et obtenons la figure 4.6. Comme on peut le voir, l'hypothèse de lognormalité semble plausible puisque les histogrammes obtenus sont en forme de cloche. Plus formellement, le test de normalité de Shapiro-Wilk (Shapiro et Wilk, 1965) est appliqué. On retrouve les p-valeurs du test sur chaque histogramme sur la figure 4.6. Le test de Shapiro-Wilk teste l'hypothèse nulle selon laquelle les données sont issues d'une population de loi normale. Au seuil critique $\alpha = 0,05$, les p-valeurs sont supérieures à α, donc l'hypothèse H_0 est acceptée.

Notons Y la variable dépendante et $X = (X_1, X_2, \ldots, X_k)^T$ le vecteur des variables explica-

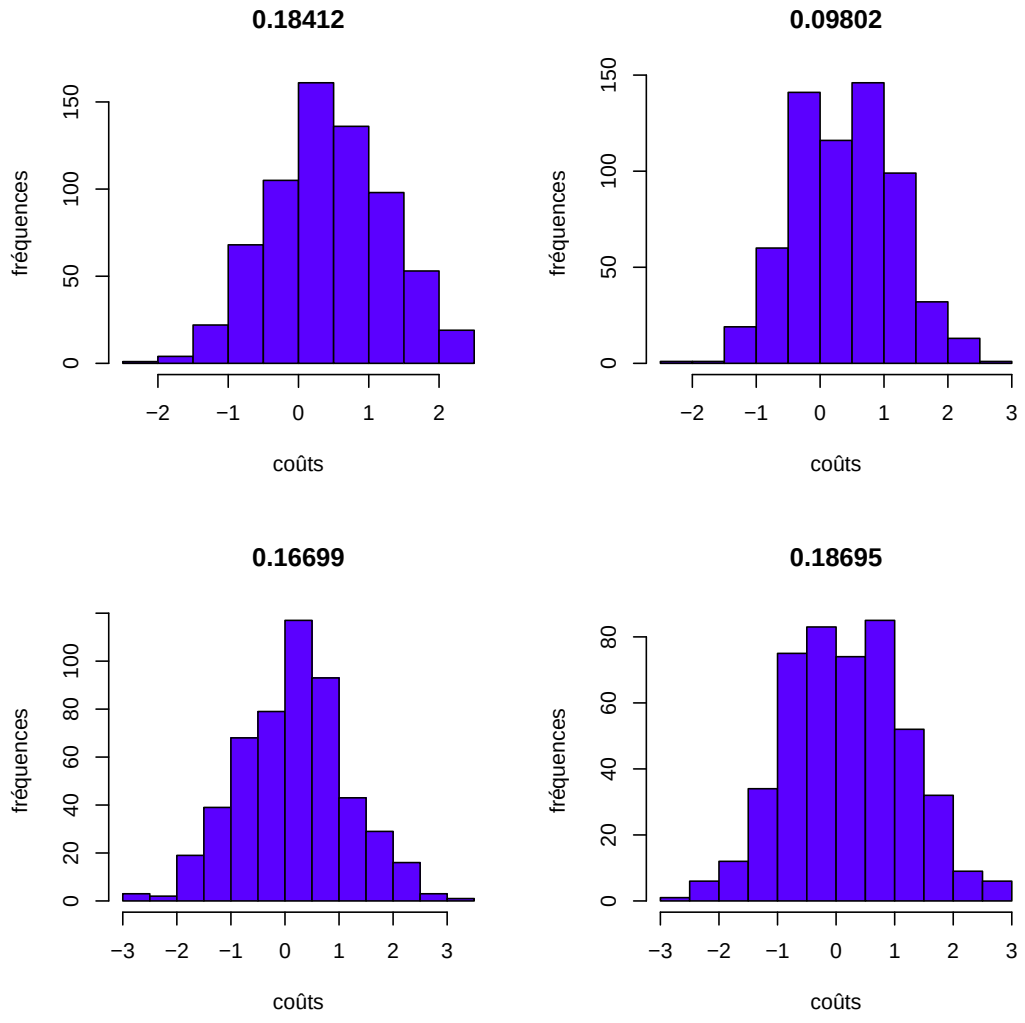

Figure 4.6 Histogramme du logarithme des coûts non nuls pour quatre jeux de données.

tives. On pose $Z = \ln(Y)$. Il s'agit alors d'ajuster simplement un modèle linéaire à la variable

Z. On a :

$$Z = X^T\beta + \epsilon,$$

où β est le vecteur des paramètres à estimer et ϵ une variable aléatoire de loi normale centrée et de variance σ^2. L'estimation de β est assez directe avec la méthode des moindres carrés ; celle-ci permet d'obtenir une expression analytique pour β.

Prédiction avec le modèle lognormal

Avec le modèle lognormal, nous obtenons des résultats pour le logarithme du coût. Or ce qui nous intéresse c'est le coût lui-même et non son logarithme. Il nous faut donc se ramener à l'échelle initiale. La méthode naturelle consiste à appliquer l'exponentielle aux prédictions puisque cette dernière est la fonction réciproque du logarithme. Dans la littérature, un certain nombre d'auteurs se contentent de cette technique. Or nous savons que l'espérance du logarithme d'une variable aléatoire est strictement inférieure au logarithme de l'espérance de cette variable. Autrement dit

$$\mathbb{E}(\ln(X)) < \ln(\mathbb{E}(X)).$$

Ainsi en prenant directement l'exponentielle, nous obtenons une prédiction biaisée. On se rappelle que nous avons fait une hypothèse de lognormalité pour la variable coût. L'espérance et la variance d'une loi lognormale sont :

$$\mu_X = \exp\left(\mu_Y + \frac{\sigma_Y^2}{2}\right) \quad \text{et} \quad \sigma_Y^2 = \mu_X^2 \exp(\sigma_Y^2 - 1).$$

On peut donc s'inspirer de l'expression de l'espérance pour effectuer les prévisions. Ainsi on a

$$\hat{y}_i = \exp(\hat{z}_i + \frac{1}{2}\hat{\sigma}_Y^2),$$

où $\hat{z}_i$ est la prédiction du modèle lognormal et $\hat{\sigma}_Y^2$ l'estimation de la variance du modèle lognormal.

4.3.2 Le modèle Gamma

Une autre approche pour modéliser les valeurs strictement positives est le modèle Gamma qui entre dans le cadre des modèles linéaires généralisés. Ici on ne transforme pas directement la variable mais plutôt sa moyenne.

La loi Gamma (équation 4.17) prend différentes formes selon la valeur du paramètre r. Par exemple, pour $r = 1$, on a une loi exponentielle. La moyenne et la variance d'une loi gamma

$G(r, \theta)$ sont :

$$\mu = \frac{r}{\theta} \quad \text{et} \quad \sigma^2 = \frac{r}{\theta^2} = \frac{\mu}{\theta}.$$

Pour la fonction *lien*, nous choisissons la fonction inverse, soit $g(\mu) = \frac{1}{\mu}$, qui est la fonction *lien canonique*. Une autre fonction *lien* également utilisée avec le modèle Gamma est la fonction logarithmique.

La figure 4.7 représente les histogrammes de la variable coût pour les valeurs non nulles.

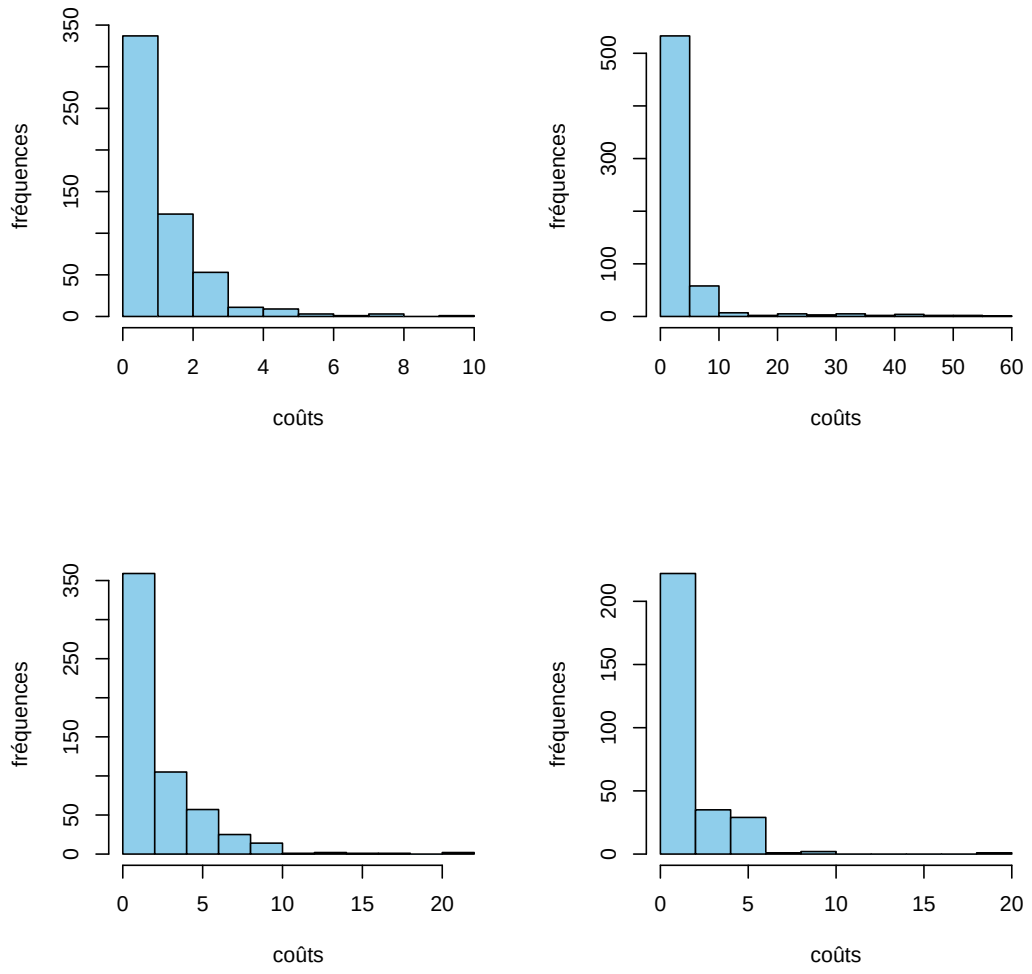

Figure 4.7 Histogramme des coûts non nuls pour quatre jeux de données.

À première vue, l'hypothèse d'une distribution exponentielle semble plausible. De plus la loi exponentielle n'étant qu'un cas particulier de la loi Gamma, il est donc tout à fait raisonnable de considérer le modèle Gamma. On peut vérifier plus formellement cette hypothèse avec

des tests d'adéquation (Kolmogorov-Smirnov, Anderson-Darling, etc) comme dans le cas de la position moyenne. Néanmoins, vu qu'une comparaison avec la méthode lognormale est effectuée ensuite, nous nous contentons des histogrammes.

Estimation des paramètres

Les deux modèles que nous considérons sont les modèles de type *hurdle* logistique-lognormal et logistique-Gamma. Notons respectivement Z et X les vecteurs des variables explicatives du modèle logistique et du modèle "strictement positif" (lognormal ou Gamma). Considérons n observations indépendantes de la forme $(y_i, \mathbf{x}_i, \mathbf{z}_i)$. Le modèle s'écrit :

$$\begin{cases} \ln\left(\frac{p_i}{1-p_i}\right) = & \mathbf{z}_i^T\gamma \\ g(\mu_i) = & \mathbf{x}_i^T\beta, \end{cases} \tag{4.28}$$

où $p_i = P(Y_i > 0|\mathbf{x}_i, \gamma)$ est la probabilité que le coût soit strictement positif, μ_i est la moyenne de la loi lognormale ou Gamma et g la fonction *lien*.

La loi lognormale ainsi que la loi Gamma ne pouvant pas générer de valeurs nulles, on a pas besoin de les tronquer comme dans le cas des données de comptage.

Les paramètres β et γ doivent être estimés. Sans surprise, c'est la méthode de maximum de vraisemblance qui est utilisée. Ici la vraisemblance se décompose en deux parties indépendantes permettant une estimation séparée des paramètres (Cameron et Trivedi, 2013). En effet la log-vraisemblance se décompose comme suit :

$$\ell(\beta, \gamma\,;\, y_1, \ldots, y_n) = \ell_1(\gamma\,;\, y_1, \ldots, y_n) + \ell_2(\beta\,;\, y_1, \ldots, y_n) \tag{4.29}$$

avec

$$\ell_1(\gamma; y_1, \ldots, y_n) = \sum_{i=1}^{n} \mathbb{1}(y_i > 0) \ln p_i + (1 - \mathbb{1}(y_i > 0)) \ln(1 - p_i) \tag{4.30}$$

et

$$\ell_2(\beta, \gamma\,;\, y_1, \ldots, y_n) = \sum_{i=1}^{n} \mathbb{1}(y_i > 0) \ln f(y_i\,;\, \beta), \tag{4.31}$$

où f est la fonction de densité de la loi lognormale ou de la loi Gamma et $\mathbb{1}(y_i > 0)$ est la fonction indicatrice qui vaut 1 si $y_i > 0$ et 0 sinon.

Les prédictions des nouvelles observations sont effectuées selon le mécanisme suivant :

$$\hat{y}_i = \begin{cases} 0 & \text{si } \hat{p}_i < p^* \\ \hat{\mu}_i & \text{sinon} \end{cases} \tag{4.32}$$

L'observation classée comme nulle par le modèle logistique est prédite à 0 et si celle-ci est prédite comme étant non nulle alors nous la prédisons par le modèle lognormal ou Gamma. Généralement le seuil de classification p^* est fixé à 0,5. Or l'excès de "zéros" dans nos données conduit à des valeurs de probabilités prédites faibles. La valeur seuil p^* permet notamment de contrôler la proportion de valeurs nulles prédites. Le choix de la valeur optimale de p^* est discuté à la section 5.3.1.

4.3.3 Choix du meilleur modèle

Les variables explicatives dont nous disposons sont le mot clé (*Keyword_id*), la position moyenne (*AveragePosition*) et les impressions (*ImpressionCount*). Nous considérons également des interactions doubles dans les modèles. La comparaison des deux modèles repose sur le choix d'un modèle lognormal ou un modèle Gamma. Notre modèle lognormal étant en réalité un modèle linéaire, nous disposons de tous les éléments classiques de vérification d'un modèle. Ainsi dans un premier temps nous vérifions la validité de ce modèle par une analyse des résidus.

Vérification du modèle linéaire (lognormal)

Nous échantillonnons des mots clés de notre grand jeu de données et créons 100 nouveaux jeux de données. Pour chacun des modèles, nous appliquons la régression logistique d'une part puis le modèle lognormal d'autre part. Normalement, il faudrait faire une validation de chaque modèle en détails. Toutefois ce travail serait très vite fastidieux. Nous choisissons donc de calculer le coefficient de détermination R^2 du modèle pour chaque jeu de données. En effet un coefficient de détermination élevé est généralement synonyme d'un modèle adéquat. Sur nos 100 jeux de données, nous obtenons en moyenne un coefficient de détermination $R^2_{\text{moyen}} = 0,607$. Ce qui est relativement raisonnable.

À titre d'illustration, la figure 4.8 présente une analyse de résidus pour un jeu de données en particulier. Le nuage de points des valeurs prédites en fonction des résidus confirme l'homoscédasticité et l'absence de tendance. Aussi, les points sont relativement alignés selon la droite de Henry sur le diagramme quantile-normal traduisant une normalité plausible des résidus. En somme le modèle lognormal semble valide.

Comparaison des deux modèles

Une analyse non exhaustive a été également faite pour le modèle Gamma pour s'assurer notamment si le modèle était globalement significatif. Un test de rapport des vraisemblances

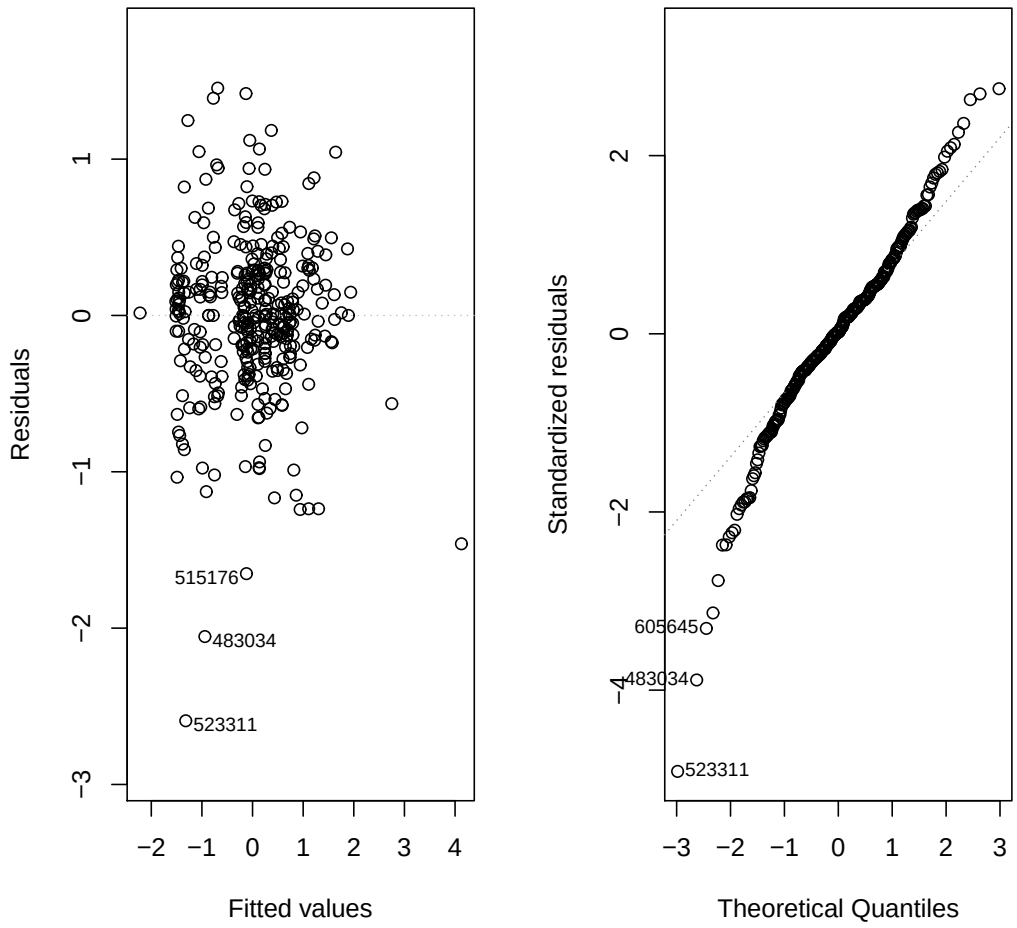

Figure 4.8 A gauche, le nuage de points des résidus en fonction des valeurs ; à droite le diagramme quantile-quantile des résidus.

semblable au test 3.13 est utilisé. L'hypothèse H_0 est toujours rejetée montrant que le modèle est significatif. Pour comparer les deux modèles, les critères d'information AIC et BIC ainsi que la racine carrée de l'erreur quadratique moyenne de prédiction sont utilisés.
Pour le modèle Gamma, le logiciel a parfois échoué à estimer les paramètres lorsque la fonction inverse est utilisée comme fonction *lien* dans le modèle. Par exemple, sur les 100 jeux de données, seuls 61 cas ont réussi avec la fonction inverse. Nous avons alors également testé la fonction *lien* logarithmique (qui elle a toujours marché). Pour nos 100 jeux de données, les deux modèles lognormal et Gamma sont ajustés ; nous calculons ensuite les critères d'information AIC et BIC ainsi que la racine carrée de l'erreur quadratique moyenne. Nous prenons

la moyenne de ces métriques pour les 100 jeux de données. Les résultats sont récapitulés dans le tableau 4.4 (code disponible à l'annexe C).

Tableau 4.4 Tableau comparatif des méthodes lognormal et Gamma.

	lognormal	Gamma (inverse)	Gamma (log)
RMSE	52,28	5,44	1347,23
AIC	905,35	845,30	1150,03
BIC	1110,92	1041,44	1355,60

Le modèle Gamma avec la fonction *lien* inverse est le meilleur selon les trois critères puisqu'il présente les plus petites valeurs. Puis vient la méthode lognormale et enfin la méthode Gamma avec le lien logarithmique. Toutefois vu que pour des raisons numériques, le modèle Gamma avec le lien inverse échoue par moment, nous gardons également le modèle lognormal. Ce dernier est utilisé en cas d'échec de la première.

Une fois les variables explicatives (position moyenne, nombre d'impressions et coût) estimées, nous appliquons notre méthode et présentons les résultats obtenus au chapitre 5.

CHAPITRE 5 PRÉSENTATION DES RÉSULTATS

Ce chapitre présente l'application de notre modèle de prédiction des taux de clics (voir section 3.2.2) sur les données à notre disposition. Notre modèle de prédiction des taux de clics consiste en une double application de la régression logistique. Aussi des variables intermédiaires sont estimées. Nous avons donc plusieurs étapes de modélisation. La position moyenne est modélisée par une loi normale tronquée à 1 (section 4.1) ; un modèle de type *hurdle* logistique et binomial négatif est ajusté au nombre d'impressions (section 4.2) et le coût est estimé par un modèle de type *hurdle* logistique et Gamma (section 4.3). Ainsi nous présentons d'abord l'algorithme global qui regroupe tous ces modèles. Ensuite nous présentons les outils utilisés dans la validation. Puis nous présentons les résultats obtenus avec notre méthode. Ces résultats sont comparés avec ceux obtenus lorsque nous considérons les données sur les trois variables explicatives comme connues, c'est-à-dire que la position moyenne, le nombre d'impressions et le coût ne sont plus estimés mais leurs valeurs sont directement utilisées dans la double régression logistique. Rappelons bien que ces données ne sont en réalité pas connues et que cette hypothèse est faite ici uniquement dans le but de comparer notre méthode.

5.1 Algorithme global de prédiction des taux de clics

L'algorithme ci-dessous présente comment les différents modèles présentés plus haut sont mis bout à bout pour, à la fin, permettre de prédire le taux de clics.

> Entrée : ensemble d'apprentissage (*train*), ensemble de test (*test*).
> L'ensemble d'apprentissage représente les données sur lesquelles les différents paramètres sont estimés tandis que l'ensemble de test, comme son nom l'indique, sert à tester la qualité des prédictions des modèles.
>
> Sortie : taux de clics prédits pour l'ensemble de test.
>
> *Les modèles sont ajustés sur l'ensemble d'apprentissage et les prédictions sont faites sur celui de test.*
>
> *I. Estimation des variables explicatives*
>
> 1. **Positions**
>   (i) détermination des paramètres des lois normales tronquées ;
>   (ii) prédiction des positions.
> 2. **Impressions**

(i) modèle de type *hurdle* Bernoulli + binomial négatif ;

(ii) prédiction des impressions.

3. **Coûts**

(i) modèle de type *hurdle* Bernoulli + Gamma ;

(ii) prédiction des coûts.
Ici il y a un choix sur le seuil de classification (modèle logistique) à faire c'est-à-dire la valeur p^* telle qu'un coût sera considéré nul si la probabilité prédite y est inférieure (voir section 4.3).

II. Approche par double régression logistique (section 3.2.2)

4. **Étiquetage des données** : il faut choisir le nombre de classes à construire pour les taux de clics compris entre 0 et 1 strictement. Ces étiquettes forment une nouvelle variable nommée *région*.

5. **Classification**

(i) modèle logistique multinomial ;

(ii) prédiction des classes.

6. **Prédiction des taux de clics**

(i) Les observations des classes "0" sont prédites à 0 et celles des classes "1" à 1.

(ii) Modèle logistique pour les observations à taux de clics compris entre 0 et 1 strictement.

(a) On retire de l'ensemble d'apprentissage les observations à taux de clics unitaire et nul ;

(b) Modèle logistique sur ce sous-ensemble d'apprentissage ;

(c) Prédiction des taux de clics.

Cet algorithme présente deux hyperparamètres qu'il faut déterminer. D'une part, dans l'étiquetage des observations à l'aide du taux de clics, au lieu de regrouper les observations à taux de clics compris entre 0 et 1 en une seule classe "2", nous pouvons les répartir en plusieurs classes. En effet, on peut effectuer un découpage de l'intervalle (0,1) et créer des sous-classes "21", "22", etc. D'autre part, dans la modélisation des coûts (voir section 4.3), il nous faut déterminer la valeur de p^* dans la phase de prédiction. Pour choisir ces hyperparamètres, nous avons besoin de métriques.

5.2 Méthodes d'évaluation

Pour vérifier la qualité de notre méthode, nous avons besoin d'outils qui nous permettent de dire si notre modèle est correct ou non. Ces outils peuvent être graphiques ou numériques.

5.2.1 Métriques de comparaison

Pour les données continues telles que les taux de clics, la métrique classique est l'erreur quadratique moyenne de prédiction (*MSE*) (voir section 4.6). Les valeurs des taux de clics sont comprises entre 0 et 1 ; ainsi l'élévation au carré donne des termes encore plus petits. Nous avons donc préféré considérer la racine carrée de cette erreur notée $RMSE = \sqrt{MSE}$ qui permet d'avoir des valeurs pas trop faibles. Comme on peut le noter, dans le cas des prédictions parfaites, le RMSE est nul. Ainsi dans l'idéal, avec notre méthode, nous espérons obtenir un RMSE petit et proche de 0.

Une autre métrique que l'on peut aussi considérer est l'erreur absolue moyenne (MAE) de prédictions définie par

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|,$$

où y_i et $\hat{y}_i$ sont respectivement les valeurs réelles et les valeurs prédites et n, la taille de l'ensemble de test. Plutôt que de sommer les carrés des différences, nous sommons la valeur absolue des différences. Tout comme pour le RMSE, la métrique est nulle pour un modèle parfait ; on veut donc qu'elle soit petite.

Une grande majorité de nos observations ayant des taux de clics nuls et parmi ceux qui sont non nuls, beaucoup ayant des taux très faibles, le modèle simple qui consiste à prédire toutes les observations par 0 a naturellement un RMSE et un MAE petit puisque la large proportion d'observations nulles sera correctement prédite. De plus, notre méthode plus sophistiquée risque d'avoir des valeurs de RMSE et MAE plus grandes parce que tous les 0 n'auront pas été correctement prédits. Pour tenir compte de cet effet, nous calculons d'une part le pourcentage P_0 de "0" correctement prédit. D'autre part, nous calculons le MAE et le RMSE uniquement pour les observations qui ont un taux de clics non nul. Dans le cas du modèle simple, nous aurons un $P_0 = 100\%$ mais un MAE et un RMSE plus élevés. Pour notre méthode, nous espérons un P_0 proche de 1 mais surtout des valeurs de MAE et de RMSE beaucoup plus faibles que celles du modèle simple. Nous avons donc deux valeurs à considérer à la fois : P_0 et RMSE ou P_0 et MAE. Ce qui n'est pas très pratique. Nous combinons alors le P_0 et la seconde métrique (MAE ou RMSE) en une nouvelle métrique Err définie comme

suit :

$$Err = (2 - P_0) \times \text{métrique}, \quad \text{métrique} \in \{MAE, RMSE\}. \tag{5.1}$$

Pour un modèle parfait, on a bien $Err = 0$ puisque $P_0 = 1$ et le MAE ou le RMSE est nul. Ainsi nous utilisons notamment cette métrique pour choisir la valeur optimale pour l'hyperparamètre nombre de classes mentionné plus haut.

5.2.2 Évaluation graphique

En plus d'outils numériques, une visualisation des résultats est souvent souhaitée. C'est pourquoi les méthodes graphiques sont très utilisées dans la vérification des modèles. Par exemple le diagramme quantile quantile permet de vérifier l'hypothèse de normalité. Ici nous utilisons également une méthode graphique. Nous traçons le nuage de points des valeurs prédites en fonction des valeurs réelles. Dans un modèle parfait, nous obtenons des points alignés le long de la première bissectrice. Ainsi pour un bon modèle, nous devons obtenir des points qui suivent cette première bissectrice sans trop s'en écarter. Ainsi, en plus des métriques, nous avons un outil de vérification supplémentaire de la qualité de nos prédictions.

5.3 Expériences et résultats

Pour rappel, nous disposons des trois jeux données A, B et C. Nous choisissons des mots clés du jeu de données C afin de créer d'autres jeux de données plus petits. Ainsi nous testons nos méthodes sur ces petits jeux de données.

D'abord nous devons déterminer la valeur des hyperparamètres que sont le nombre de classes pour les observations à taux de clics compris entre 0 et 1 et la valeur de p^*. Nous commençons d'abord par ce dernier.

5.3.1 Discussion de l'hyperparamètre p^*

Dans la modélisation des coûts, nous ajustons premièrement un modèle logistique qui permet de déterminer si un coût est nul ou non. Le modèle logistique prédit alors des probabilités. Généralement, on fixe un seuil à 0,5 c'est-à-dire qu'une observation sera classée dans une classe si la probabilité prédite est inférieure à 0,5 et dans l'autre classe sinon. Le choix de cette valeur suppose que les deux classes sont équilibrées dans l'ensemble d'apprentissage, c'est-à-dire que nous avons autant d'observations pour l'une que pour l'autre. Or dans notre cas, nous n'avons pas cet équilibre.

De plus après quelques tests sur nos données, on note que la médiane des probabilités prédites

est aux alentours de 0,1 avec pour certains jeux de données une prédiction maximale inférieure à 0,5. On remarque donc que si nous choisissons un p^* élevé alors nous prédisons beaucoup de 0 tandis qu'avec une valeur faible, nous en prédisons moins.
La proportion de 0 prédite est très importante car elle se retrouve dans la prédiction des taux de clics. En effet, lors de l'étape de classification dans le modèle de prédiction des taux de clics, nous avons une relation forte entre la variable de réponse *région* (qui représente les classes) et la variable *coût*. En effet toutes les observations prédites à coût nul ont automatiquement un taux de clic également nul. Or nous avons des faux positifs parmi ces valeurs nulles prédites.

Nous considérons un jeu de données auquel nous appliquons notre modèle global pour des valeurs de p^* de 0,05, 0,1, 0,25 et 0,5. Nous traçons alors les nuages de points des valeurs prédites en fonction des valeurs réelles. Nous obtenons la figure 5.1 qui permet d'observer l'effet du paramètre p^* sur la prédiction des taux de clics. En noir (points carrés), nous avons nos valeurs prédites et en rouge (points ronds) nous avons les valeurs réelles qui suivent naturellement la première bissectrice. L'effet du paramètre p^* se voit au niveau de l'origine. Les points noirs d'abscisse 0 représentent les prédictions des observations avec un taux de clic nul. Tous les points d'ordonnée strictement positives sont des prédictions erronées et plus l'ordonnée est élevée plus la prédiction est mauvaise. Plus la valeur de p^* augmente plus leur nombre baisse puisque plus de 0 sont prédits à l'étape des coûts. A contrario, le nombre de points le long de la droite des abscisses augmentent avec p^* ; ce qui signifie que des observations non nulles sont prédites à 0. Il y a donc un compromis à trouver.

La valeur de p^* peut être fixée en discutant avec les professionnels du domaine. En effet, quelle erreur ces personnes considèrent-elles la plus grave ? Es-ce que l'on veut s'assurer qu'une observation à taux de clics nul soit bien prédite à 0 ; dans ce cas une valeur élevée de p^* est nécessaire. Ou alors on ne veut surtout pas passer à côté d'une observation qui a un taux de clic non nul ; dans quel cas une petite valeur de p^* est préférable. Malheureusement ne disposant pas de ces personnes ressources, nous optons pour une autre méthode.

Nous souhaitons prédire le maximum d'observations nulles possibles sans toutefois avoir trop de faux positifs. En gros nous souhaitons maximiser le rappel et la précision (voir section 3.1.3). Pour ce faire nous utilisons la F-mesure qui est la moyenne harmonique de ces deux quantités. La F-mesure admet un paramètre α (équation 3.12) qui permet d'accorder plus d'importance à une l'une ou l'autre des quantités. Dans notre cas, de grandes valeurs de p^* donnent un rappel élevé. Ainsi si nous donnons la même importance aux deux termes, nous obtenons des p^* grands. Nous décidons (arbitrairement) donc d'accorder trois fois plus d'importance à la précision en fixant donc la valeur de α à $1/3$.

La valeur optimale p^*_{opt} de p^* est donc celle qui maximise la quantité $F_{1/3}$. Nous considérons

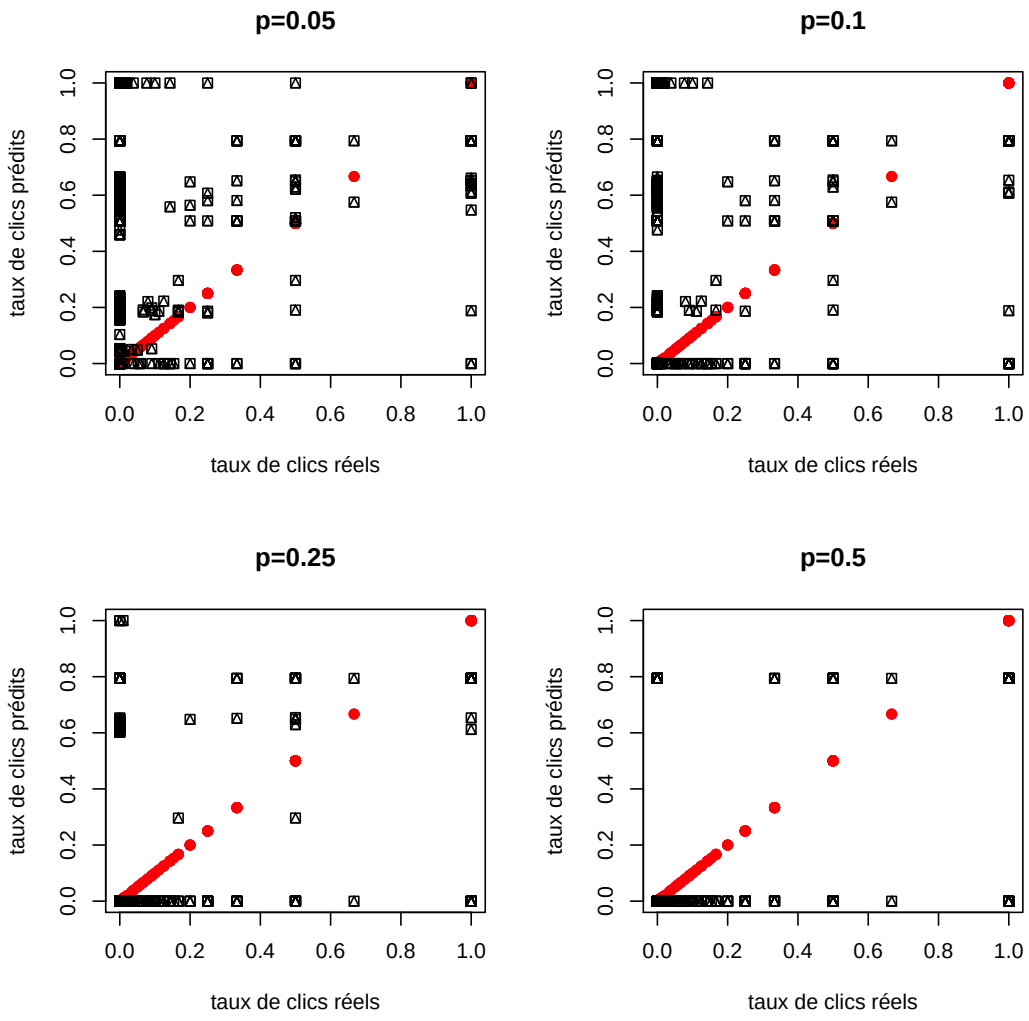

Figure 5.1 Nuage de points des taux de clics prédits en fonction des taux de clics réels pour différentes valeurs de p^*, le seuil de classification du modèle logistique des coûts pour un même jeu de données.

des valeurs de p^* comprises entre 0,05 et 0,5 avec un pas de 0,025. Avec cette méthode, nous pouvons maintenant calculer la valeur optimale de p^* pour chaque jeu de données en utilisant une méthode de validation croisée de type *10-fold cross validation*.

La validation croisée est assez coûteuse en temps de calcul puisque nous devons ajuster le modèle 10 fois. Ainsi nous avons cherché à voir si nous pouvons considérer une même valeur quelque soit le jeu de données. Nous considérons donc 50 jeux de données créés à partir du jeu de données C, puis nous calculons la valeur de p^* optimale. Le tableau ci-dessous récapitule les valeurs obtenues.

Tableau 5.1 Tableau récapitulatif des valeurs optimales de p^* obtenues sur les 50 jeux de données.

min	1er quartile	médiane	moyenne	3ème quartile	max
0,1	0,125	0,1375	0,14	0,1875	0,2

On note des valeurs comprises entre 0,1 et 0,2 avec une médiane à 0,1375 et une moyenne de 0,14. Dans la suite de nos expériences, nous considérons $p^* = 0,1375$ pour tous les jeux de données.
L'hyperparamètre p^* fixé, il nous reste à déterminer le nombre de classes à utiliser dans notre modèle.

5.3.2 Choix du nombre de classes

Nous créons 50 jeux de données du grand jeu de données C. Chaque jeu de données contient 30 mots clés suivis sur 90 jours. Nous considérons un nombre de classes entre 1 et 4 pour les observations à taux de clics compris entre 0 et 1. Pour chaque nombre de classes, nous ajustons notre modèle sur les 50 jeux de données et nous calculons différentes métriques : le MAE, le RMSE, la proportion P_0, l'erreur Err ainsi que le taux de mauvaise classification (*taux miscl*) des classes. Nous faisons ensuite la moyenne de ces métriques. Nous avons également appliqué notre modèle en utilisant les données réelles des positions moyennes, des impressions et des coûts, c'est-à-dire qu'on suppose les données sur ces variables connues et ces dernières ne sont plus estimées. Les étapes 1, 2 et 3 de la modélisation en « chaîne » (présentée à la page 25) sont donc omises. L'étape 4 reste inchangée, c'est-à-dire que la double régression logistique est encore utilisée. Cette deuxième méthode nous permet d'évaluer l'effet des différentes prédictions intermédiaires. Le tableau 5.2 présente l'ensemble des résultats obtenus pour une, deux, trois et quatre classes.

Tableau 5.2 Comparaison de notre modèle selon le choix du nombre de classes pour les taux de clics compris entre 0 et 1 strictement.

	application de notre modèle				utilisation des données réelles			
Métrique	nombre de classes				nombre de classes			
	1	2	3	4	1	2	3	4
taux miscl	0,11496	0,11320	0,13739	0,14089	0,00298	0,00835	0,01368	0,01866
P_0	0,94905	0,95042	0,94769	0,94919	0,99972	0,99979	0,99974	0,99930
RMSE	0,21423	0,21143	0,20514	0,21855	0,13079	0,11807	0,10496	0,11033
MAE	0,14610	0,13871	0,14917	0,15378	0,08923	0,07430	0,06805	0,07213
$(2 - P_0) \times$ RMSE	0,22515	0,22192	0,21588	0,22965	0,13083	0,11810	0,10499	0,11041
$(2 - P_0) \times$ MAE	0,15354	0,14558	0,15698	0,16159	0,08926	0,07432	0,06807	0,07218

D'abord comme on pouvait s'y attendre, on remarque que le taux de mauvaise classification augmente en fonction du nombre de classes. Aussi lorsque nous utilisons les données réelles, nous obtenons une classification quasi parfaite tandis qu'avec notre modèle ce taux est beaucoup plus important. Ensuite en regardant P_0, on voit que plus de 94% des zéros prédits sont des vrais zéros. En utilisant les valeurs réelles, encore une fois, nous avons une prédiction presque parfaite des zéros. Puis pour le RMSE et le MAE, nous obtenons avec notre modèle environ le double des valeurs obtenus en utilisant les valeurs réelles ; nous avons la même chose avec la métrique Err (équation 5.1) qui utilise le P_0 et les métriques RMSE et MAE. Selon cette métrique avec le RMSE, le nombre optimal de classes est trois contre deux lorsque nous considérons le P_0 et le MAE. Pour les valeurs réelles, trois est le nombre de classes optimal. Finalement nous choisissons deux classes car c'est également ce nombre de classes qui présente le plus faible taux de mauvaise classification avec notre modèle.

En somme nous choisissons de regrouper les observations à taux de clics compris entre 0 et 1 strictement en deux classes.

Jusqu'ici nous n'avons pas explicité comment nous créions nos classes. Nous utilisons les quantiles des taux de clics des observations comprises entre 0 et 1. Ainsi pour deux classes, nous utilisons la médiane : les taux de clics inférieurs à la médiane forment une classe "21" et ceux supérieurs forment la seconde classe "22". De même pour trois et quatre classes. Ce découpage nous permet d'avoir des classes à effectifs équilibrés. D'autres découpages pourraient être proposés.

La figure 5.2 présente le nuage de points des taux de clics prédits en fonction des taux de clics réels pour le jeu de données A. Nous ajustons les modèles sur les trois premiers mois et

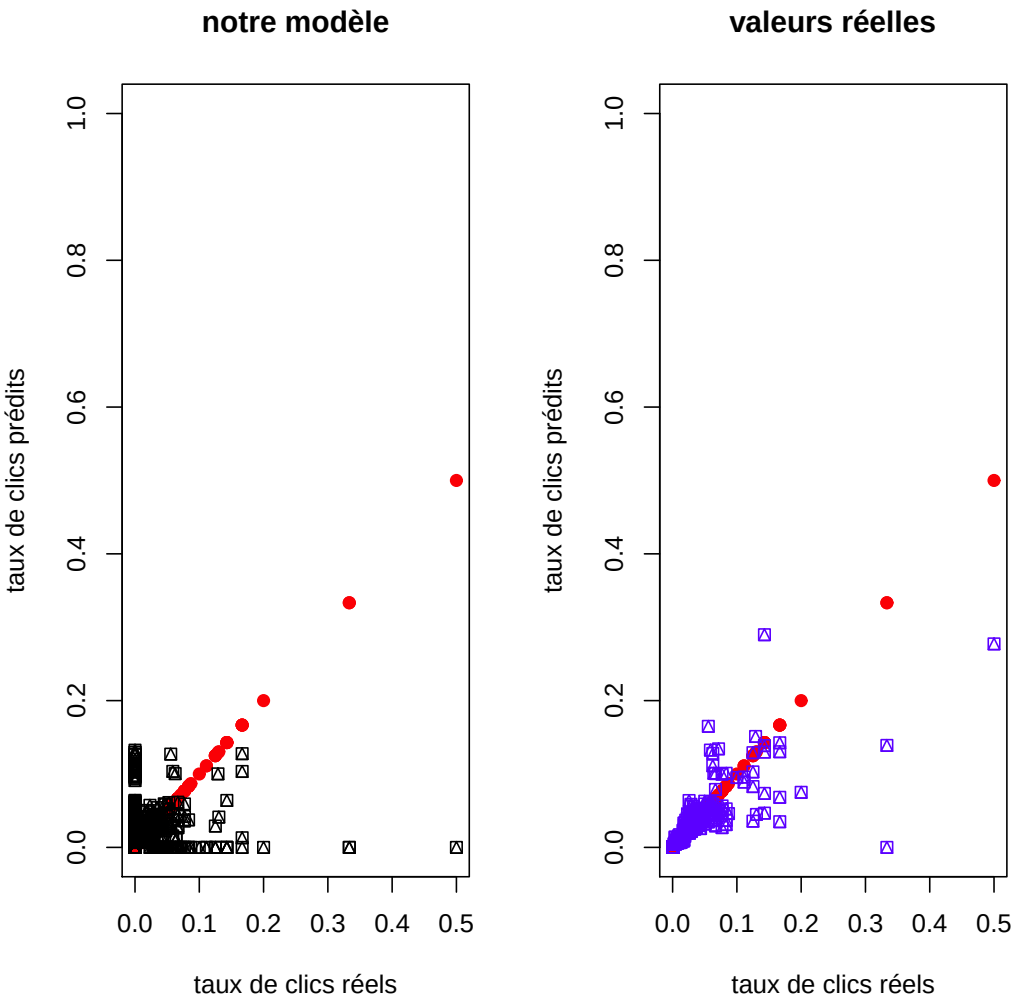

Figure 5.2 Nuage de points des taux de clics prédits en fonction des taux de clics réels sur le jeu de données A. À gauche, les résultats de notre modèle ; à droite ceux obtenus avec les données réelles.

prédisons le quatrième. La valeur de $p^* = 0,15$ optimale a été trouvée par validation croisée et nous avons choisi deux classes. Comme on peut le voir, nos prédictions suivent sensiblement la première bissectrice. Pour les petites valeurs de taux de clics, nous restons très proches de la première bissectrice. En effet les valeurs prédites sont très proches des valeurs réelles. Par contre pour les grandes valeurs de taux de clics, nos prédictions s'écartent plus de la première bissectrice. Nous expliquons ce phénomène par le faible nombre d'observations pour ces valeurs. Ce qui entraîne une plus grande variabilité dans les prédictions. Le même effet s'observe également lorsque nous utilisons les données réelles (figure 5.2).

CHAPITRE 6 CONCLUSION

6.1 Synthèse des travaux

Internet est aujourd'hui le premier espace publicitaire. En effet, les grandes entreprises ainsi que les petites investissent de plus en plus dans la publicité en ligne. Ce marché est principalement gouverné par le taux de clics qui oriente d'un côté les stratégies des annonceurs dans l'optimisation de leurs campagnes. De l'autre les revenus des moteurs de recherche et leurs algorithmes dépendent de la connaissance de ce taux de clics.

Dans ce présent mémoire, nous proposons de prédire le taux de clics en utilisant en variables explicatives, les mots clés comme variable catégorique, la position moyenne des mots clés, le nombre d'impressions et le coût des clics. Les données des annonces textuelles sont caractérisées par un fort pourcentage de données à taux de clics nul. Nous utilisons un classificateur pour d'une part écarter ces données et d'autre part un modèle logistique pour prédire les taux de clics des données restantes.

Aussi nous montrons que dans nos données, les observations sont indépendantes. En effet, malgré les mesures répétées sur les mots clés, l'agrégation journalière semble rompre la dépendance temporelle. Ensuite, avec l'historique des positions et les mots clés, nous sommes capables d'estimer les prochaines positions des mots clés. Puis avec le modèle de type *hurdle* logistique et binomial négatif, nous estimons les impressions. Notons que dans ce modèle, nous utilisons également les positions moyenne comme variable explicative. Avec la position et les impressions, nous estimons la variable coût avec un modèle de type *hurdle* également, mais avec une régression Gamma pour la deuxième étape. Finalement tous ces modèles sont mis ensemble pour fournir une procédure pour prédire les taux de clics à partir des mots clés et d'un historique de leurs réalisations. Pour appliquer la première régression logistique comme classificateur, nous créons quatre classes en utilisant le taux de clics.

Nos résultats sont comparés avec le modèle qui suppose la connaissance des données des variables explicatives. En utilisant les différentes métriques (RMSE, MAE et *Err*), nous obtenons environ le double des valeurs avec notre modèle. Les valeurs obtenues sur ces métriques sont à relativiser car elles permettent avant tout de comparer des modèles entre eux. Ainsi obtenir le double des valeurs signifie simplement que notre modèle est moins performant que celui avec les données sur les variables connues mais ne remet pas totalement en question sa qualité. De plus le graphique des valeurs prédites en fonction des valeurs réelles montre que nos prédictions sont acceptables surtout pour les petites valeurs de CTR. Le faible nombre

d'observations pour les grandes valeurs de CTR ne permet pas de bien apprendre cette région.

Le modèle que nous proposons a la particularité d'être assez simple dans la mesure où son ajustement ne nécessite qu'un historique des variables explicatives. En effet, les informations sur les utilisateurs, les annonceurs et les annonces elles-mêmes ne sont pas utilisées dans le modèle. Aussi ce modèle peut être facilement enrichi avec la connaissance d'informations supplémentaires telles que le max CPC par exemple. Enfin, des données géographiques ou encore des données sur les utilisateurs ne peuvent qu'enrichir le modèle. Même si notre modèle fonctionne plutôt bien, il présente quelques limites.

6.2 Limitations de la solution proposée

Une campagne publicitaire compte généralement un grand nombre de mots clés. Par exemple les jeux de données A et B contiennent respectivement 33 et 65 mots clés. Pour notre modèle, ces valeurs restent raisonnables. Toutefois si ce nombre devient grand, quelques problèmes peuvent se poser. Dans les différentes modélisations (position, impressions et coût), les mots clés sont utilisés comme une variable catégorique. Plus le nombre de catégories est élevée, plus le risque que certaines catégories soient similaires, c'est-à-dire que des mots clés aient un comportement identique est élevé. On a alors une multicolinéarité dans les données qui entraîne l'échec de l'estimation de certains paramètres. La solution à ce problème consiste à faire une sélection de variables c'est-à-dire retirer les catégories qui entraînent la colinéarité. Dans ce contexte, cette sélection n'est pas évidente. Cependant une méthode simple est la régularisation : il s'agit d'ajouter un terme de pénalité dans l'expression de la vraisemblance à optimiser. On dispose de la régression LASSO (Least Absolute Shrinkage and Selection Operator) basée sur la norme ℓ_1 et la régularisation Ridge sur la norme ℓ_2 (James et al., 2013). Leur mise en place n'est pas trop compliquée. En R, il existe des packages qui les proposent.
On voit donc que notre modèle peut être amélioré. Ci-dessous, nous proposons des pistes supplémentaires à explorer.

6.3 Améliorations futures

Dans le tableau 5.2 qui présente l'évolution des différentes métriques en fonction du nombre de classes (pour les observations compris entre 0 et 1 strictement), on note surtout le grand écart entre le taux de mauvaise classification avec notre modèle et celui qui suppose les données sur les variables explicatives connues. En effet nous obtenons des taux de l'ordre de 10%

contre à peine 1% lorsque les données sont connues. Ainsi la principale amélioration à apporter réside à ce niveau. Nous avons utilisé le modèle logistique comme classificateur, nous pensons que des méthodes plus sophistiquées permettraient d'obtenir de meilleurs résultats. Notons toutefois que ni les arbres de classification ni l'analyse discriminante linéaire n'ont donné de meilleurs résultats. Nous pensons que les machines à vecteur de support (SVM, Support Vector Machine) et éventuellement les réseaux de neurones sont des alternatives qui peuvent donner de meilleurs résultats.

Notre modélisation de la variable position moyenne peut également être améliorée. Actuellement elle repose uniquement sur les mots clés et les positions réalisées. Avec le nouveau jeu de données C et son grand nombre de campagnes, on peut essayer d'exploiter de l'information des variables catégoriques. L'idée de Baqapuri et Trofimov (2014) d'utiliser des réseaux de neurones pour modéliser les variables catégoriques nous semble être une solution possible. En effet aujourd'hui les réseaux de neurones ont montré leur capacité à surpasser les méthodes d'apprentissage classique dans différentes tâches. Ainsi on peut envisager utiliser un tel modèle pour les positions.

En somme nous avons une modélisation en chaîne inédite pour le taux de clics. Au vu de nos résultats corrects, il est évident que d'autres améliorations sont encore possibles.

RÉFÉRENCES

L. Adjengue, *Méthodes statistiques : concepts, applications et exercices ; Luc Adjengue*. Montréal : Presses internationales Polytechnique, 2014.

E. Agichtein, E. Brill, S. Dumais, et R. Ragno, "Learning user interaction models for predicting web search result preferences", dans *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2006, pp. 3–10.

K. Aho, D. Derryberry, et T. Peterson, "Model selection for ecologists : the worldviews of aic and bic", *Ecology*, vol. 95, no. 3, pp. 631–636, 2014.

P. D. Allison, "Measures of fit for logistic regression". SAS Global Forum, Washington, DC, 2014. En ligne : `https://support.sas.com/resources/papers/proceedings14/1485-2014.pdf`

R. Altman, "Lecture 26 : Models for gamma data", 2009. En ligne : `http://people.stat.sfu.ca/~raltman/stat402/402L26.pdf`

N. Archak, V. Mirrokni, et S. Muthukrishnan, "Budget optimization for online advertising campaigns with carryover effects", dans *Sixth Ad Auctions Workshop*, 2010.

C. Assari, "Classification de mots-clés des campagnes publicitaires sur les moteurs de recherche et calcul de prévisions", Mémoire de maîtrise, École Polytechnique de Montréal, août 2014.

A. I. Baqapuri et I. Trofimov, "Using neural networks for click prediction of sponsored search", *arXiv preprint arXiv :1412.6601*, 2014.

D. R. Barr et E. T. Sherrill, "Mean and variance of truncated normal distributions", *The American Statistician*, vol. 53, no. 4, pp. 357–361, 1999.

C. Borgs, J. Chayes, N. Immorlica, K. Jain, O. Etesami, et M. Mahdian, "Dynamics of bid optimization in online advertisement auctions", dans *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 531–540.

P. J. Brockwell et R. A. Davis, *Time series : theory and methods*. Springer Science & Business Media, 2013.

J. Burkardt, “The truncated normal distribution”, *Department of Scientific Computing Website*, 2014.

C. Bys, “La publicité en ligne dépasse la pub tv, mais les annonceurs sont inquiets”, Janvier 2017, [consulté le 07 Juin 2017].

A. C. Cameron et P. K. Trivedi, *Regression analysis of count data.* Cambridge university press, 2013, vol. 53.

O. Chapelle, E. Manavoglu, et R. Rosales, “Simple and scalable response prediction for display advertising”, *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 4, p. 61, 2015.

A. Charpentier, “Notes de cours”, 2013. En ligne : `http://freakonometrics.free.fr/slides-2040-4.pdf`

N. Craswell, O. Zoeter, M. Taylor, et B. Ramsey, “An experimental comparison of click position-bias models”, dans *Proceedings of the 2008 International Conference on Web Search and Data Mining.* ACM, 2008, pp. 87–94.

B. DasGupta et S. Muthukrishnan, “Stochastic budget optimization in internet advertising”, *Algorithmica*, pp. 1–28, 2013.

D. A. Dickey et W. A. Fuller, “Distribution of the estimators for autoregressive time series with a unit root”, *Journal of the American statistical association*, vol. 74, no. 366a, pp. 427–431, 1979.

P. Duchesne, “Séries chronologique univariées, notes de cours”, 2016. En ligne : `http://www.dms.umontreal.ca/~duchesne/stt6615.html`

J. Faraway, G. Marsaglia, J. Marsaglia, et A. Baddeley, *goftest : Classical Goodness-of-Fit Tests for Univariate Distributions*, 2017, r package version 1.1-1. En ligne : `https://CRAN.R-project.org/package=goftest`

J. Friedman, T. Hastie, et R. Tibshirani, *The elements of statistical learning.* Springer series in statistics Springer, Berlin, 2001, vol. 1.

O. Gaudoin, “Statistique inférentielle avancée, notes de cours”, 2013. En ligne : `http://www-ljk.imag.fr/membres/Olivier.Gaudoin/SIA.pdf`

Google AdWords, “Glossaire - aide adwords”, 2017, [consulté le 07 juin 2017]. En ligne : https://support.google.com/adwords/topic/3121777?hl=fr&ref_topic=3119071

T. Graepel, J. Q. Candela, T. Borchert, et R. Herbrich, “Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft’s bing search engine”, dans *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 13–20.

J. T. Hattaway, “Parameter estimation and hypothesis testing for the truncated normal distribution with applications to introductory statistics grades”, 2010.

D. W. Hosmer, S. Lemeshow, et R. X. Sturdivant, *Applied logistic regression*. John Wiley & Sons, 2013, vol. 398.

S. Jackman, *pscl : Classes and Methods for R Developed in the Political Science Computational Laboratory, Stanford University*, Department of Political Science, Stanford University, Stanford, California, 2015, r package version 1.4.9. En ligne : http://pscl.stanford.edu/

C. H. Jackson, “Multi-state models for panel data : The msm package for R”, *Journal of Statistical Software*, vol. 38, no. 8, pp. 1–29, 2011. En ligne : http://www.jstatsoft.org/v38/i08/

G. James, D. Witten, T. Hastie, et R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, vol. 6.

Z. Jiang, “Research on ctr prediction for contextual advertising based on deep architecture model”, *Journal of Control Engineering and Applied Informatics*, vol. 18, no. 1, pp. 11–19, 2016.

T. Joachims, L. Granka, B. Pan, H. Hembrooke, et G. Gay, “Accurately interpreting click-through data as implicit feedback”, dans *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. Acm, 2005, pp. 154–161.

R. Kumar, S. M. Naik, V. D. Naik, S. Shiralli, V. Sunil, et M. Husain, “Predicting clicks : Ctr estimation of advertisements using logistic regression classifier”, dans *Advance Computing Conference (IACC), 2015 IEEE International*. IEEE, 2015, pp. 1134–1138.

D. Kwiatkowski, P. C. Phillips, P. Schmidt, et Y. Shin, “Testing the null hypothesis of

stationarity against the alternative of a unit root : How sure are we that economic time series have a unit root ?" *Journal of econometrics*, vol. 54, no. 1-3, pp. 159–178, 1992.

D. Lambert, "Zero-inflated poisson regression, with an application to defects in manufacturing", *Technometrics*, vol. 34, no. 1, pp. 1–14, 1992.

J. Lee, Y. Shi, F. Wang, H. Lee, et H. K. Kim, "Advertisement clicking prediction by using multiple criteria mathematical programming", *World Wide Web*, vol. 19, no. 4, pp. 707–724, 2016.

K.-c. Lee, B. B. Orten, A. Dasdan, et W. Li, "Estimating conversion rate in display advertising from past performance data", Août 13 2012, uS Patent App. 13/584,545.

M. Lefebvre, *Processus stochastiques appliqués : Mario Lefebvre*, deuxième édition. – éd. Montréal : Presses internationales Polytechnique, 2014.

L'encyclopédie illustrée du marketing, "Définition : Moteur de recherche", 2017, [consulté le 07 juin 2017]. En ligne : `https://www.definitions-marketing.com/definition/moteur-de-recherche-2/`

P. McCullagh et J. A. Nelder, "Generalized linear models, no. 37 in monograph on statistics and applied probability", 1989.

H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin *et al.*, "Ad click prediction : a view from the trenches", dans *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 1222–1230.

NetMarketShare. (2017) Desktop search engine market share. [consulté le 07 juin 2017]. En ligne : `https://www.netmarketshare.com/search-engine-market-share.aspx?qprid=4&qpcustomd=0`

J. Nocedal et S. J. Wright, *Sequential quadratic programming*. Springer, 2006.

PwC, "Internet advertising", 2016, [consulté le 07 Juin 2017].

P. Quinn, "Modélisation et prédiction du comportement de mots-clés dans des campagnes publicitaires sur les moteurs de recherche", Mémoire de maîtrise, École Polytechnique de Montréal, avril 2011.

R. Rakotomalala, “Pratique de la régression logistique”, *Régression Logistique Binaire et Polytomique, Université Lumière Lyon*, vol. 2, 2011.

M. Richardson, E. Dominowska, et R. Ragno, “Predicting clicks : estimating the click-through rate for new ads”, dans *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 521–530.

Y. Sasaki *et al.*, “The truth of the f-measure”, *Teach Tutor mater*, vol. 1, no. 5, 2007.

S. S. Shapiro et M. B. Wilk, “An analysis of variance test for normality (complete samples)”, *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965.

R. H. Shumway et D. S. Stoffer, *Time series analysis and its applications : with R examples*. Springer Science & Business Media, 2010.

I. Trofimov, A. Kornetova, et V. Topinskiy, “Using boosted trees for click-through rate prediction for sponsored search”, dans *Proceedings of the Sixth International Workshop on Data Mining for Online Advertising and Internet Economy*. ACM, 2012, p. 2.

UIT, “Ict facts and figures 2016”, juillet 2016.

K. J. van Garderen, “Optimal prediction in loglinear models”, *Journal of econometrics*, vol. 104, no. 1, pp. 119–140, 2001.

W. N. Venables et B. D. Ripley, *Modern Applied Statistics with S*, 4e éd. New York : Springer, 2002, iSBN 0-387-95457-0. En ligne : `http://www.stats.ox.ac.uk/pub/MASS4`

J. M. Ver Hoef et P. L. Boveng, “Quasi-poisson vs. negative binomial regression : How should we model overdispersed count data ?” *Ecology*, vol. 88, no. 11, pp. 2766–2772, 2007.

Wikipédia, “Loi tronquée — wikipédia, l’encyclopédie libre”, 2016, [En ligne ; Page disponible le 2-mai-2017]. En ligne : `http://fr.wikipedia.org/w/index.php?title=Loi_tronqu%C3%A9e&oldid=131999138`

Wikipedia, “Multinomial logistic regression — wikipedia, the free encyclopedia”, 2017, [Online ; accessed 29-April-2017].

Wikipedia, “Logistic regression — wikipedia, the free encyclopedia”, 2017, [Online ; accessed 29-April-2017].

C. J. Wu, “On the convergence properties of the em algorithm”, *The Annals of statistics*, pp. 95–103, 1983.

A. Zeileis, C. Kleiber, et S. Jackman, “Regression models for count data in R”, *Journal of Statistical Software*, vol. 27, no. 8, 2008. En ligne : http://www.jstatsoft.org/v27/i08/

A. Zeileis, C. Kleiber, et S. Jackman, “Regression models for count data in r”, *Journal of statistical software*, vol. 27, no. 8, pp. 1–25, 2008.

Y. Zhou, D. Chakrabarty, et R. Lukose, “Budget constrained bidding in keyword auctions and online knapsack problems”, dans *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008, pp. 1243–1244.

Z. A. Zhu, W. Chen, T. Minka, C. Zhu, et Z. Chen, “A novel click model and its applications to online advertising”, dans *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 2010, pp. 321–330.

A. F. Zuur, E. N. Ieno, N. J. Walker, A. A. Saveliev, et G. M. Smith, *Mixed effects models and extensions in ecology with R*. New York, NY : Springer New York, 2009. En ligne : http://dx.doi.org/10.1007/978-0-387-87458-6_11

ANNEXE A LA POSITION MOYENNE : CODE R

Nous utilisons la fonction **fitdistr()** du package **MASS** (Venables et Ripley, 2002) pour calculer les estimateurs de maximum de vraisemblance. Le package **msm** (Jackson, 2011) fournit la distribution de la loi normale tronquée.
Certains mots clés ont des positions qui ne varient pas du tout ou très peu (toujours la même valeur). Nous obtenons alors soit un écart type nul ou très petit. Dans ce cas, nous prédisons simplement une valeur constante ou la moyenne dans le cas de l'écart type très petit. Le code R ci-dessous présente la fonction **pos_param** qui prend l'ensemble d'apprentissage en entrée et estime les moyennes et les variances pour chaque mot clé.

```
require(MASS)
require(msm)
pos_param <- function(train) {
  # fonction de densité de la loi normale tronquée
  dtnorm0 <- function(x, mean, sd) {
    dtnorm(x, mean, sd, lower = 1)
  }
  estimator <- matrix(0, nrow=nlevels(train$KEYWORD_ID), ncol=2)
  j <- 1
  for (i in levels(train$KEYWORD_ID)) {
    # on récupére les positions passées du mot clé i
    tmp <- train[train$KEYWORD_ID==i,]
    # valeurs initiales pour la maximisation
    mu <- mean(tmp$AVERAGEPOSITION)
    s <- sd(tmp$AVERAGEPOSITION)
    # estimation des paramétres
    if (s!=0) {
      est.par <- try(fitdistr(tmp$AVERAGEPOSITION, densfun = dtnorm0,
             start=list(mean=mu, sd=s), control=list(maxit=1000)))
      if(inherits(est.par, "try-error")) {
        estimator[j,] <- c(mu, s)
        j <- j + 1
        next
      }
      estimator[j,] <- estimator.par$estimate
    } else {
      estimator[j,] <- c(mu, 0)
    }
```

```
    j <- j + 1
  }
  rownames(estimator) <- levels(train$KEYWORD_ID)
  estimator
}
```

La fonction **predict_pos()** permet de prédire les positions sur l'ensemble de test.

```
predict_pos <- function(test, estimator) {
  pstn <- function(i) {
    if(estimator[i,2]==0)
      estimator[i,1]
    else
      rtnorm(1, mean = estimator[i,1], sd=estimator[i, 2], lower = 1)
  }
  sapply(test$KEYWORD_ID, function(i) pstn(i))
}
```

Les fonctions **cvm.test()** et **ad.test()** du package **goftest** (Faraway et al., 2017) permettent respectivement d'effectuer les tests d'adéquation de Cramer-von Mises et d'Anderson-Darling. Le code ci-dessous effectuent ces tests et donne le nombre de fois où l'hypothèse H_0 est rejetée.

```
require(goftest)
test_adequation = matrix(NA, nrow=nlevels(train$KEYWORD_ID), ncol=2)
j <- 1
for (i in levels(train$KEYWORD_ID)) {
  tmp <- train$KEYWORD_ID==i
  test_adequation[j, 1] <- cvm.test(train$AVERAGEPOSITION[tmp], "ptnorm", mean=estimator
      [j,1],
                  sd=estimator[j,2], lower=1)$p.value
  test_adequation[j, 2] <- ad.test(train$AVERAGEPOSITION[tmp], "ptnorm", mean=estimator[
      j,1],
                 sd=estimator[j,2], lower=1)$p.value
  j <- j + 1
}
cat('CVM : H0 est rejetée ', sum(test_adequation[,1]<0.05), ' fois sur ',
  nlevels(train$KEYWORD_ID), '\n')
cat('AD : H0 est rejetée ', sum(test_adequation[,2]<0.05), ' fois sur ',
  nlevels(train$KEYWORD_ID), '\n')
```

ANNEXE B LES IMPRESSIONS : CODE R

La fonction **glm()** de R permet d'ajuster les modèles Poisson et Quasi-Poisson. Le package **pscl** (Jackman, 2015; Zeileis et al., 2008b) contient les fonctions **hurdle()** et **zeroinfl** pour les modèles de type *hurdle* et *zero inflated*. Le code R ci-dessous permet d'obtenir les valeurs du tableau 4.2. Les variables explicatives sont la position et les mots clés dans tous les modèles Toutefois dans les modèles *zero inflated*, une des deux variables a été retirée car des problèmes numériques survenaient telles que la singularité de matrices.

```
require(MASS)
require(pscl)
reg_pois <- glm(IMPRESSIONCOUNT~AVERAGEPOSITION+KEYWORD_ID, data=jeuA, family='poisson')
reg_qpois <- glm(IMPRESSIONCOUNT~AVERAGEPOSITION+KEYWORD_ID, data=jeuA,
               family='quasipoisson')
reg_nb <- glm.nb(IMPRESSIONCOUNT~AVERAGEPOSITION+KEYWORD_ID, data=jeuA)
zinb <- zeroinfl(I(IMPRESSIONCOUNT-1)~AVERAGEPOSITION + KEYWORD_ID | AVERAGEPOSITION,
               dist="negbin", data=jeuA)
zip <- zeroinfl(I(IMPRESSIONCOUNT-1)~KEYWORD_ID| AVERAGEPOSITION + KEYWORD_ID,
              dist="poisson", data=jeuA)
hnb <- hurdle(I(IMPRESSIONCOUNT-1)~AVERAGEPOSITION+KEYWORD_ID, dist="negbin", data=jeuA)
hpois <- hurdle(I(IMPRESSIONCOUNT-1)~AVERAGEPOSITION+KEYWORD_ID, dist="poisson",
              data=jeuA)

fm <- list("Poisson" = reg_pois, "NB" = reg_nb, "ZINB" = zinb, "ZIP" = zip,
          "HNB" = hnb, "HP" = hpois)
print(rbind(Df = sapply(fm, function(x) attr(logLik(x), "df")),
           AIC = sapply(fm, function(x) AIC(x)),
           BIC = sapply(fm, function(x) -2*logLik(x) + attr(logLik(x), "df")*log(nrow(
               jeuA)))))
```

ANNEXE C LE COÛT : CODE R

Le code R ci-dessous permet d'obtenir le tableau 4.4.

```
# key_index : les indices des mots clés
# keys : ensemble des mots clés
# jeuC : jeu de données C
set.seed(12345)
ctrl <- glm.control(maxit = 250, epsilon = 1e-6, trace=F)
rm <- matrix(NA, 100, 3)
ai <- matrix(NA, 100, 3)
bi <- matrix(NA, 100, 3)
for (l in 1:100) {
  k <- sample(key_index, 30)
  dt <- jeuC[jeuC$KEYWORD_ID %in% names(keys)[k],]
  dt <- dt[dt$COSTUSD!=0,]  # on conserve uniquement les couts différents de 0
  dt$KEYWORD_ID <- droplevels(dt$KEYWORD_ID)

  # modéle lognormal
  cost_lm <- lm(log(COSTUSD)~KEYWORD_ID:IMPRESSIONCOUNT+IMPRESSIONCOUNT + KEYWORD_ID,
      data=dt)
  rm[l,1] <- rmse(exp(predict(cost_lm, newdata=dt) + 0.5*summary(cost_lm)$sigma**2),dt$
      COSTUSD)
  ai[l,1] <- AIC(cost_lm)
  bi[l,1] <- BIC(cost_lm)
  # gamma inverse
  gamma_inv = try(glm(COSTUSD~KEYWORD_ID:IMPRESSIONCOUNT+IMPRESSIONCOUNT + KEYWORD_ID,
      data=dt, family = Gamma(link="inverse"), control = ctrl))
  if(!inherits(gamma_inv, "try-error")) {
    rm[l,2] <- rmse(predict(gamma_inv, newdata=dt, type='response'), dt$COSTUSD)
    ai[l,2] <- AIC(gamma_inv)
    bi[l,2] <- BIC(gamma_inv)
  }
  # gamma logartithme
  gamma_log <- glm(COSTUSD~KEYWORD_ID:IMPRESSIONCOUNT+IMPRESSIONCOUNT + KEYWORD_ID,data=
      dt, family = Gamma(link="log"), control = ctrl)
  rm[l,3] <- rmse(predict(gamma_log, newdata=dt, type='response'), dt$COSTUSD)
  ai[l,3] <- AIC(gamma_log)
  bi[l,3] <- BIC(gamma_log)
}
fm <- list("RMSE" = rm, "AIC" = ai, "BIC" = bi)
```

```
res <- t(sapply(fm, function(x) sapply(1:3, function(i) mean(na.omit(x[,i])))))
colnames(res) <- c("Lognormal", "Gamma (inverse)", "Gamma (log)")
res
```

ANNEXE D EXPÉRIENCES ET RÉSULTATS : CODE R

Le code ci-dessous permet la mise en application du modèle de prédiction des taux de clics présenté dans ce mémoire. Ce script prend les données d'apprentissage en entrée et estime les taux de clics sur l'ensemble de test. Les variables explicatives sont également estimées. De plus un graphique des valeurs prédites en fonction des vraies valeurs est également tracé. Ce script utilise les fonctions **ajust()**, **threshold() balancedata()** dont le code est disponible plus bas. La première permet de retirer les niveaux non utilisés dans la variable *KEYWORD_ID*. La deuxième calcule la valeur optimale de p^* (voir équation 4.32). La fonction **balancedata()** quant à elle s'assure que tous les mots clés soient présents dans les ensembles d'apprentissage.

```
require(MASS)
require(nnet)
require(msm)
require(pscl)
# parametre pour la fonction glm
ctrl <- glm.control(maxit = 250, epsilon = 1e-6, trace=F)

# étiquetage des données et création de la variable région
train$region <- NA
train$region[train$CTR==0] <- 0
ctr_null <- train$CTR!=0 & train$CTR!=1
med <- median(train$CTR[ctr_null])
train$region[ctr_null] <- sapply(which(ctr_null==T), function(i) ifelse(train$CTR[i]>med
    , 3, 2))
train$region[train$CTR==1] <- 1

#### Positions ####
estimateurs <- pos_param(train)
test$AVERAGEPOSITION <- predict_pos(test, estimateurs)

#### Impressions ####
hnb <- try(hurdle(I(IMPRESSIONCOUNT-1)~AVERAGEPOSITION+KEYWORD_ID | poly(AVERAGEPOSITION
    , 2) +KEYWORD_ID, dist="negbin", data=train))
# en cas d'échec, on retire la position du modele binomial négatif
if(inherits(hur, "try-error")) {
  hnb <- hurdle(I(IMPRESSIONCOUNT-1)~KEYWORD_ID | poly(AVERAGEPOSITION, 2) + KEYWORD_ID,
       dist="negbin", data=train)
}
```

```
impr_preds = predict(hnb, type="response", newdata = test)
test$IMPRESSIONCOUNT <- floor(impr_preds + 1)

#### Couts ####
train$cout_binaire <- ifelse(train$COSTUSD==0, 0, 1)
cost_log <- glm(cout_binaire~KEYWORD_ID + IMPRESSIONCOUNT + AVERAGEPOSITION +
    IMPRESSIONCOUNT:KEYWORD_ID + AVERAGEPOSITION:IMPRESSIONCOUNT,data=train, family=
    binomial, control=ctrl)
cost_preds <- predict(cost_log, newdata=test, type='response')
cost_id <- cost_preds>threshlod(train) # identifiant des coéts non nuls
# on récupére les données écoét non nul
cost_train <- ajust(train[train$COSTUSD!=0,])
cost_test <- ajust(test[cost_id,])

cost_gamma <- try(glm(COSTUSD~KEYWORD_ID:IMPRESSIONCOUNT+IMPRESSIONCOUNT + KEYWORD_ID,
data=cost_train, family = Gamma(link="inverse"), control = ctrl))
if(!inherits(cost_gamma, "try-error")) {
  test$COSTUSD[cost_id] <- predict(cost_gamma, newdata=cost_test, type='response')
} else {
# en cas d'échec, du lien inverse pour le modéle gamma, le modéle lognormal est ajusté
  cost_lm <- lm(log(COSTUSD)~KEYWORD_ID:IMPRESSIONCOUNT+IMPRESSIONCOUNT + KEYWORD_ID,
data=cost_train)
  test$COSTUSD[cost_id] <- exp(predict(cost_lm, newdata=cost_test) + 0.5*summary(cost_lm
      )$sigma**2)
}
test$COSTUSD[!cost_id] = 0

#### on retire les observations a CTR=0 ou CTR=1 ####
# 1er modéle logistique
classificateur <- multinom(region~IMPRESSIONCOUNT*COSTUSD*AVERAGEPOSITION + KEYWORD_ID +
     KEYWORD_ID:IMPRESSIONCOUNT,
data=train, maxit=2000, reltol=1e-5, trace = F)
clss <- predict(classificateur, newdata=test)
# new_train contient les observations étaux de clics compris entre 0 et 1
new_train <- ajust(train[train$region!=1 & train$region!=0,])
new_test <- ajust(test[clss!=0 & clss!=1,])
new_train <- balancedata(train, new_train, levels(new_test$KEYWORD_ID))

# 2éme modéle logistique
fit <- glm(cbind(CLICKCOUNTM, IMPRESSIONCOUNT-CLICKCOUNTM)~COSTUSD+AVERAGEPOSITION +
    KEYWORD_ID + factor(region),
data=new_train, family = "binomial", control = ctrl)
preds <- numeric(nrow(test))
```

```
preds[clss==0] <- 0
predf[clss==1] <- 1
if(nrow(new_test)!=0) {
  new_test$region <- clss[clss!=0 & clss!=1]
  new_test <- ajust(new_test)
  preds[clss!=0 & clss!=1] <- predict(fit, newdata=new_test, type='response')
}
# ctr contient les vraies valeurs de CTR
plot(ctr, ctr, col='red', pch=19, xlab = "taux de clics réels", ylab = "taux de clics pr
    édits")
points(ctr, preds, pch=14, col='blue')
```

```
ajust <- function(data) {
  data$KEYWORD_ID = droplevels(data$KEYWORD_ID)
  data
}
```

```
threshlod <- function(train) {
  # calcul de la F mesure
  fmesure <- function(preds, reels) {
    mat_conf <- table(reels, preds)
    prec <- mat_conf[1,1]/sum(mat_conf[,1])
    sens <- mat_conf[1,1]/sum(mat_conf[1,])
    b <- 1/3
    (1+b**2)*prec*sens/(b**2*prec + sens)
  }
  n <- nrow(train)
  probs <- seq(0.05,0.5, by = 0.025)
  res <- numeric(length(probs))
  idx <- sample(n, n)
  nfold <- nrow(train)%/%10
  i <- 1
  # 10 fold validation
  for(p in probs) {
    tmp <- 0
    for (k in 0:9) {
      if(k==9)
        sub <- idx[(nfold*k+1):n]
```

```
      else
        sub <- idx[(nfold*k+1):(nfold*(k+1))]
      fit <- glm(cout_binaire~KEYWORD_ID + IMPRESSIONCOUNT + AVERAGEPOSITION +
          IMPRESSIONCOUNT:KEYWORD_ID + AVERAGEPOSITION:IMPRESSIONCOUNT,data=train,
          family=binomial, subset = -sub)
      preds <- predict(fit, newdata=train[sub,], type='response')
      preds <- ifelse(preds>p, 1, 0)
      tmp <- tmp + fmesure(preds, train$co[sub])
    }
    res[i] <- tmp/10
  i <- i + 1
  }
  probs[which.max(res)]
}
```

```
balancedata<-function(data, train, test_levels) {
  a = levels(train$KEYWORD_ID)
  for (j in test_levels) {
    if(!any(j==a))
      train = rbind(train, data[data$KEYWORD_ID==j,])
  }
  train = ajust(train)
  train
}
```